

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

MICROSOFT CORPORATION,
Petitioner,

v.

QOMPLX LLC,
Patent Owner.

IPR2026-00183
Patent: USP 11,539,663
Issued: December 27, 2022
Application No. 16/412,340
Filed: May 14, 2019

Title: SYSTEM AND METHOD FOR
MIDSERVER FACILITATION OF LONG-HAUL
TRANSPORT OF TELEMETRY FOR CLOUD-BASED SERVICES

PETITION FOR
INTER PARTES REVIEW OF U.S. PATENT NO. 11,539,663

TABLE OF CONTENTS

	Page
LIST OF EXHIBITS	v
MANDATORY NOTICES UNDER 37 C.F.R. § 42.8	vii
1. Real Party-In-Interest.....	vii
2. Related Matters	vii
3. Lead And Back-Up Counsel, And Service Information.....	vii
I. INTRODUCTION	1
II. GROUNDS FOR STANDING PER 37 C.F.R. § 42.104(a)	2
III. IDENTIFICATION OF CHALLENGE	2
A. Statement Of The Precise Relief Requested / Statutory Grounds.....	2
IV. STATE OF THE ART AND ORDINARY SKILL IN THE ART	3
A. State Of The Art	3
1. Virtual Appliances Were Commonly Used For Installation Of Applications On Virtual Machines	3
2. Open Virtual Appliances (OVAs) Were A Known And Standard Format For Automatically Installing And Configuring Applications In Virtual Machines	7
3. Midservers Were Well-Known And Facilitated Data Transfer Between Local Networks And Cloud Platforms	10
B. Person Of Ordinary Skill In The Art.....	15
V. THE '663 PATENT	16
A. The Claims' Earliest Possible Priority Date Is May 14, 2019	16
B. The '663 Patent's Specification	16
C. The Prosecution History	18
D. The Challenged Claims	20

VI.	CLAIM CONSTRUCTION	21
VII.	GROUND 1: CLAIMS 1 AND 4 ARE OBVIOUS OVER MATURANA IN VIEW OF KASHYAP	21
A.	Maturana (EX1004).....	22
B.	Kashyap (EX1005).....	25
C.	Combining Maturana And Kashyap.....	28
D.	Claim 1	37
1.	Element [1.1]	37
2.	Element [1.2].....	39
3.	Element [1.3].....	42
4.	Element [1.4].....	48
5.	Element [1.5].....	50
6.	Element [1.6].....	53
7.	Element [1.7].....	57
E.	Claim 4	58
1.	Element [4.1].....	58
2.	Element [4.2].....	59
3.	Element [4.3].....	60
4.	Element [4.4].....	60
5.	Element [4.5].....	60
6.	Element [4.6].....	61
7.	Element [4.7].....	61
VIII.	NO OBJECTIVE INDICIA OF NON-OBVIOUSNESS	61
IX.	CONCLUSION.....	61
	CERTIFICATE OF COMPLIANCE.....	63

CERTIFICATE OF SERVICE 64

TABLE OF AUTHORITIES

Page

Cases

<i>Fonar Corp. v. Gen. Elec. Co.</i> , 107 F.3d 1543 (Fed. Cir. 1997)	36
<i>Keynetik, Inc. v. Samsung Elecs. Co.</i> , No. 2022-1127, 2023 WL 2003932 (Fed. Cir. Feb. 15, 2023)	36
<i>Phillips v. AWH Corp.</i> , 415 F.3d 1303 (Fed. Cir. 2005)	21

Statutes

35 U.S.C. § 103	3
-----------------------	---

Rules

37 C.F.R. § 42.104	2
37 C.F.R. § 42.24	63
37 C.F.R. § 42.6	64
83 Fed. Reg. 51340	21

LIST OF EXHIBITS

No.	Description
1001	U.S. Patent No. 11,539,663 (“ 663 patent ”)
1002	File History of U.S. Patent No. 11,539,663
1003	Declaration of Mark Crovella, dated Jan. 13, 2026 (“ Crovella Decl. ” or “ Crovella ”)
1004	U.S. Patent App. Pub. No. 2015/0281355 A1 to Maturana et al. (“ Maturana ”)
1005	U.S. Patent App. Pub. No. 2012/0246639 A1 to Kashyap et al. (“ Kashyap ”)
1006	U.S. Patent App. Pub. No. 2015/0178113 A1 to Steven Dake (“ Dake ”)
1007	B. Sosinsky, <i>Cloud Computing Bible</i> (2011) (“ Sosinsky ”) (excerpts)
1008	U.S. Patent App. Pub. No. 2012/0216181 A1 to Arcese et al. (“ Arcese ”)
1009	U.S. Patent App. Pub. No. 2013/0152080 A1 to Sachindran et al. (“ Sachindran ”)
1010	X. Jin et al., <i>Reinventing Virtual Appliances</i> , IBM J. of Res. and Dev., 53(4), Paper 7 (2009) (“ Jin ”)
1011	S. Mastrianni et al., <i>IT Autopilot: A Flexible IT Service Management and Delivery Platform for Small and Medium Business</i> , IBM Sys. J., 46(3), at 609-24 (2007) (“ Mastrianni ”)
1012	U.S. Patent App. Pub. No. 2017/0366606 A1 to Ben-Shaul et al. (“ Ben-Shaul ”)
1013	U.S. Patent App. Pub. No. 2016/0048408 A1 to Madhu et al. (“ Madhu ”)
1014	Gabor Kecskemeti, et al., <i>Toward Efficient Appliance Delivery With Minimal Manageable Virtual Appliances</i> , IEEE Trans. On Services Computing, 7(2), at 279-92 (2014) (“ Kecskemeti ”)

No.	Description
1015	U.S. Patent No. 10,579,403 to Antony et al. (“ Antony ”)
1016	U.S. Patent App. Pub. No. 2007/0294676 A1 to Mellor et al. (“ Mellor ”)
1017	U.S. Patent App. Pub. No. 2019/0026138 A1 to Cao et al. (“ Cao ”)
1018	U.S. Patent App. Pub. No. 2019/0065213 A1 to Li et al. (“ Li ”)
1019	DMTF, <i>Open Virtualization Format White Paper</i> , ver. 1.0.0 (2009) (“ OVF-2009 ”)
1020	DMTF, <i>Open Virtualization Format White Paper</i> , ver. 2.0.0 (2014) (“ OVF-2014 ”)
1021	DMTF, <i>Open Virtualization Format Specification</i> , ver. 2.1.1 (2015) (“ OVF-2015 ”)
1022	U.S. Patent App. Pub. No. 2013/0283364 A1 to Chang et al. (“ Chang ”)
1023	Plaintiff Qomplx’s Preliminary Infringement Contentions (“ PICs ”)
1024	List of Challenged Claims

MANDATORY NOTICES UNDER 37 C.F.R. § 42.8

1. Real Party-In-Interest

Microsoft Corporation is the sole real party-in-interest.

2. Related Matters

The '663 patent (EX1001) has been asserted against Microsoft in the following district court litigation (the “district court case”):

- *Qomplx LLC v. Microsoft Corporation*, Case No. 1-25-cv-01383 (W.D. Tex.), filed August 28, 2025.

The '663 patent has also been asserted in the following district court litigation:

- *Qomplx LLC v. Palo Alto Networks, Inc.*, Case No. 2-25-cv-00913 (E.D. Tex.), filed August 28, 2025.

3. Lead And Back-Up Counsel, And Service Information

Lead Counsel	Back-up Counsel
Andrew M. Mason, Reg. No. 64,034 andrew.mason@klarquist.com	Frank Morton-Park, Reg. No. 80,750 frank.morton-park@klarquist.com Todd M. Siegel, Reg. No. 73,232 todd.siegel@klarquist.com Samuel B. Thacker, Reg. No. 78,633 samuel.thacker@klarquist.com
KLARQUIST SPARKMAN, LLP 121 SW Salmon Street, Suite 1600 Portland, Oregon, 97204 503-595-5300 (phone) 503-595-5301 (fax)	

Petitioner consents to service via email at the above email addresses and the email address of Msft-Qomplx@klarquist.com.

Pursuant to 37 C.F.R. § 42.10(b), concurrently filed with this Petition is a Power of Attorney executed by Petitioner and appointing the above counsel.

Petitioner authorizes Account No. 02-4550 to be charged for any fees, including those enumerated in 37 C.F.R. § 42.15.

I. INTRODUCTION

Microsoft Corporation (“Petitioner”) respectfully requests *inter partes* review (“IPR”) of claims 1 and 4 of U.S. Patent No. 11,539,663 (“the ’663 patent”) (EX1001), allegedly owned by QOMPLX LLC (“Patent Owner”). For the reasons set forth below, the Board should institute IPR and hold these claims unpatentable.

The challenged claims relate to ingestion of data into a cloud-based service, where a midserver receives data from a plurality of computing devices, applies a plurality of transformations to the received data, and retransmits the received data over a secure connection. Each feature was well-known and conventional and, years before the priority date, Maturana (EX1004) had described the combination of features in the challenged claims.

Maturana, which was not before the Office during original prosecution, discloses an on-premise cloud agent application that “facilitates collection of industrial data from one or more data sources” and “migration of the collected data to a cloud platform for storage and processing.” EX1004, Abstract. Maturana further teaches transforming the data “before uploading to the cloud, rather than requiring this transformation to be performed on the cloud.” EX1004, [0055]. Kashyap (EX1005), in turn, expressly teaches how to efficiently implement cloud computing applications using the well-known Open Virtual Application (“OVA”) standard. The combination of Maturana and Kashyap renders each challenged claim obvious.

The challenged claims were allowed after the Applicant heavily amended independent claims 1 and 4 to include additional claim elements and remove original claim elements. Prior to allowance, the Examiner did not adapt the search to address the new claim elements but instead simply repeated the same search queries used for the original claims. Ultimately, neither Maturana nor Kashyap were considered during original prosecution. The Board should institute *inter partes* review and evaluate the patentability of the claims against these important prior art references.

II. GROUND FOR STANDING PER 37 C.F.R. § 42.104(a)

Petitioner certifies that the '663 patent is available for IPR and that Petitioner is not barred or estopped from requesting an IPR challenging the patent claims on the grounds identified in this petition.

III. IDENTIFICATION OF CHALLENGE

A. Statement Of The Precise Relief Requested / Statutory Grounds

Petitioner requests *inter partes* review of claims 1 and 4 (the “challenged claims”) of the '663 patent, on the following statutory ground:

	Reference(s)	Basis	Claims
Ground 1	Maturana (EX1004) and Kashyap (EX1005)	35 U.S.C. § 103	1, 4

For this ground, in Section VII below, the petition presents evidence of unpatentability and establishes a reasonable likelihood that the Petitioner will prevail in establishing that each challenged claim is unpatentable.

IV. STATE OF THE ART AND ORDINARY SKILL IN THE ART

The following sections explain pertinent aspects of the state of the art and the level of ordinary skill in the art as of May 14, 2019, the filing date of the application that led to challenged claims and earliest effective invention date. *Infra* Sections IV.A – V.A. The state of the art and level of ordinary skill would be the same even if the challenged claims were entitled to an effective invention date of April 28, 2016, the filing date of the ancestor application to which the '663 patent alleges CIP priority. EX1003, ¶26.

A. State Of The Art

1. Virtual Appliances Were Commonly Used For Installation Of Applications On Virtual Machines

Virtual appliances went hand in hand with virtual machines. A virtual appliance was known in the art as a virtual machine image containing a pre-installed operating system and one or more pre-installed software applications, such that the

software applications run when loaded in a virtual machine. EX1003, ¶30. As just one example, a 2012 patent publication explains this in its background section:

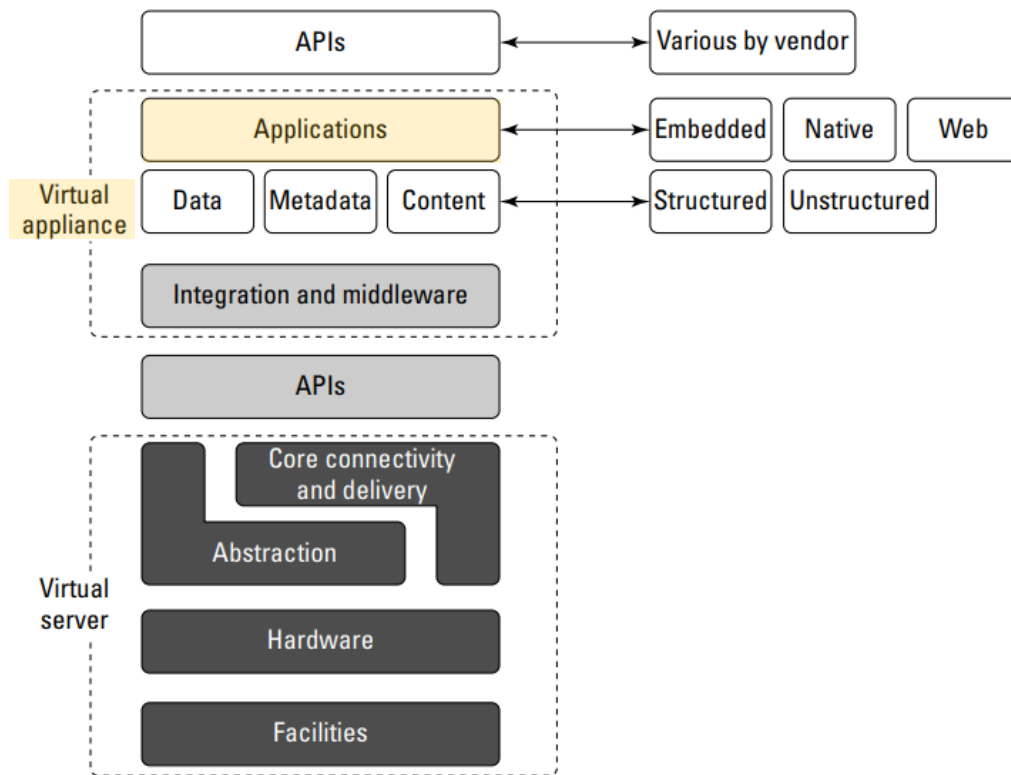
A virtual appliance is a virtual machine image file that includes a pre-configured operating system and a single application. It is packaged as a pre-installed system image for a particular virtualized environment and hardware platform. A common file format for the virtual appliance is the open virtualization format (OVF). ... Accordingly, the virtual appliance is a software image containing an application designed to run inside a virtual machine.

EX1005 (Kashyap), [0003]; *see also* EX1016 (Mellor), [0002] (explaining the benefits of providing applications as a virtual appliance, which can be “administered as black boxes”); EX1003, ¶30.

By 2011, virtual appliances were already regarded as “a very important standard cloud computing deployment object” (EX1007, 45), as explained in Wiley Publishing’s “Cloud Computing Bible” by Sosinsky (*id.*, a, e, xi). This is because virtual appliances provide the software applications comprising a cloud platform and thus “occupy the middle of the cloud computing stack” (*id.*, 50-51), as shown in Sosinsky Figure 3.2:

FIGURE 3.2

A virtual appliance is software that installs as middleware onto a virtual machine.



EX1007, 50-51 (highlighting added); EX1003, ¶31.

Sosinsky explained that “[o]ne of the major advantages of a virtual appliance is that you can use the appliances as the basis for assembling more complex services, the appliance being one of your standardized components.” EX1007, 51. Virtual appliances thus enable a modular plug-and-play / “ready-to-use” approach for adding software functionality. EX1007, 52-55 (showing and discussing vendors providing virtual appliances); EX1003, ¶32.

POSITAs knew that “instantiating virtual appliances ... that encapsulate the required applications or services (the *key functionality*) with their support

environment (e.g., an operating system and several software libraries)” was a standard technique for quickly and easily deploying software applications. EX1014 (Kecskemeti), 279 (emphasis original). Virtual appliances enable rapid provisioning because they “simplify and automate service deployment and eliminate the complexities in the service installation process.” EX1010 (Jin), 5; *id.*, 2 (explaining that virtual appliances simplify deployment and “eliminate the need for manual configuration and relieve the management burden”), 4-5 (discussing “complex and labor-intensive process” for manually deploying applications without virtual appliances); *see also* EX1007, 52 (“Virtual appliances are far easier to install and run than an application that you must set up yourself.”); EX1003, ¶33.

It was well-known to use a bootstrap process, or boot process, to install and configure a virtual appliance automatically upon instantiation, so that a base virtual appliance can be deployed and then customized / configured for that specific instance. EX1006 (Dake), [0009]-[0010], [0018]-[0023]; *see also* EX1013 (Madhu), [0127], [0129] (explaining that an initial “bootstrap process involves the automatic deployment, creation, and use of ... virtual infrastructure,” which begins when “a virtual machine [] is downloaded to an on-premise data center”); EX1016, [0019] (explaining how information provided by a virtual appliance is used to “instantiate[]” / “install” the virtual appliance on a virtual machine monitor /

hypervisor), [0027] (explaining that a virtual appliance may be “responsible for its own configuration and modification”); EX1003, ¶34.

**2. Open Virtual Appliances (OVAs) Were A Known
And Standard Format For Automatically Installing
And Configuring Applications In Virtual Machines**

Virtual appliances were such a standard tool for provisioning cloud resources that the Distributed Management Task Force (DMTF), an industry working group of over 160 companies and organizations, developed the Open Virtualization Format (“OVF”) “for the packaging and distribution of virtual appliances composed of one or more virtual computer systems.” EX1019, 3; EX1020, 8 (“OVF is a packaging format for virtual appliances.”); EX1007, 55 (OVF is the “best known” file format for virtual appliances), 107 (noting that OVF is an open standard supported by “notable virtualization vendors”), 243-44 (OVF is “a standard method for creating, packaging, and provisioning virtual appliances”). EX1003, ¶35.

The resulting OVF software package was referred to as an Open Virtual Appliance or “OVA.” EX1021, 14 (“The file extension [for the OVF package] is .ova (open virtual appliance or application)”); *see also* EX1020, 14; EX1016 (Mellor), [0002]-[0003] (2007 patent describing an “Open Virtual Appliance (OVA)”); EX1017, [0001] (explaining that “OVA (open virtual appliance) template file is a single archive file in TAR file format that contains files for portability and

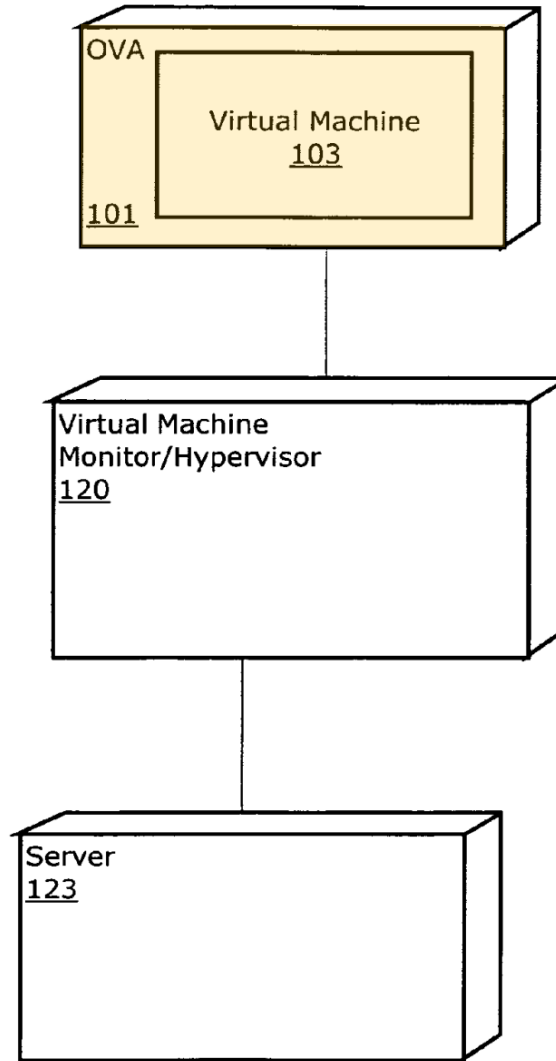
deployment of virtualization appliances,” where the OVA template file includes “an OVF (open virtual format) descriptor file”); EX1003, ¶36.¹

OVAAs are packaged with enough metadata “to automatically and securely install, configure, and run the virtual appliance on any virtualization platform.” EX1019, 5; EX1020, 8. OVF thus provides “full support for *unattended installation* to ease the burden of deployment for users, and thereby enhance the user’s experience.” EX1019, 8 (emphasis added); EX1020, 10 (same). EX1003, ¶37.

A 2007 patent application described an “Open Virtual Appliance (OVA)” as a “packaging format for virtual machines 103 (VMs) or virtual appliances.” EX1016, [0019]. As shown in FIG. 1 below, a “virtual appliance may include one or more virtual machines 103,” and the VM 103 “may include an application and an operating system.” EX1016, [0019].

¹ The ’663 patent itself describes—without any detail or explanation—implementing the midserver with an “Open Virtual Appliance (OVA).” EX1001, 16:16-21.

Figure 1



EX1016, FIG. 1 (highlighting added). The OVA is “instantiated on any virtual platform,” such as the virtual machine monitor (VMM) / hypervisor 120. Further, “[t]he VMM 120 allocates the server’s 123 resources that the VM requires for operation.” EX1016, [0019]. An OVA is provided as “an executable instruction set (e.g., an xml file), including VM specific information (i.e. metadata), for directing

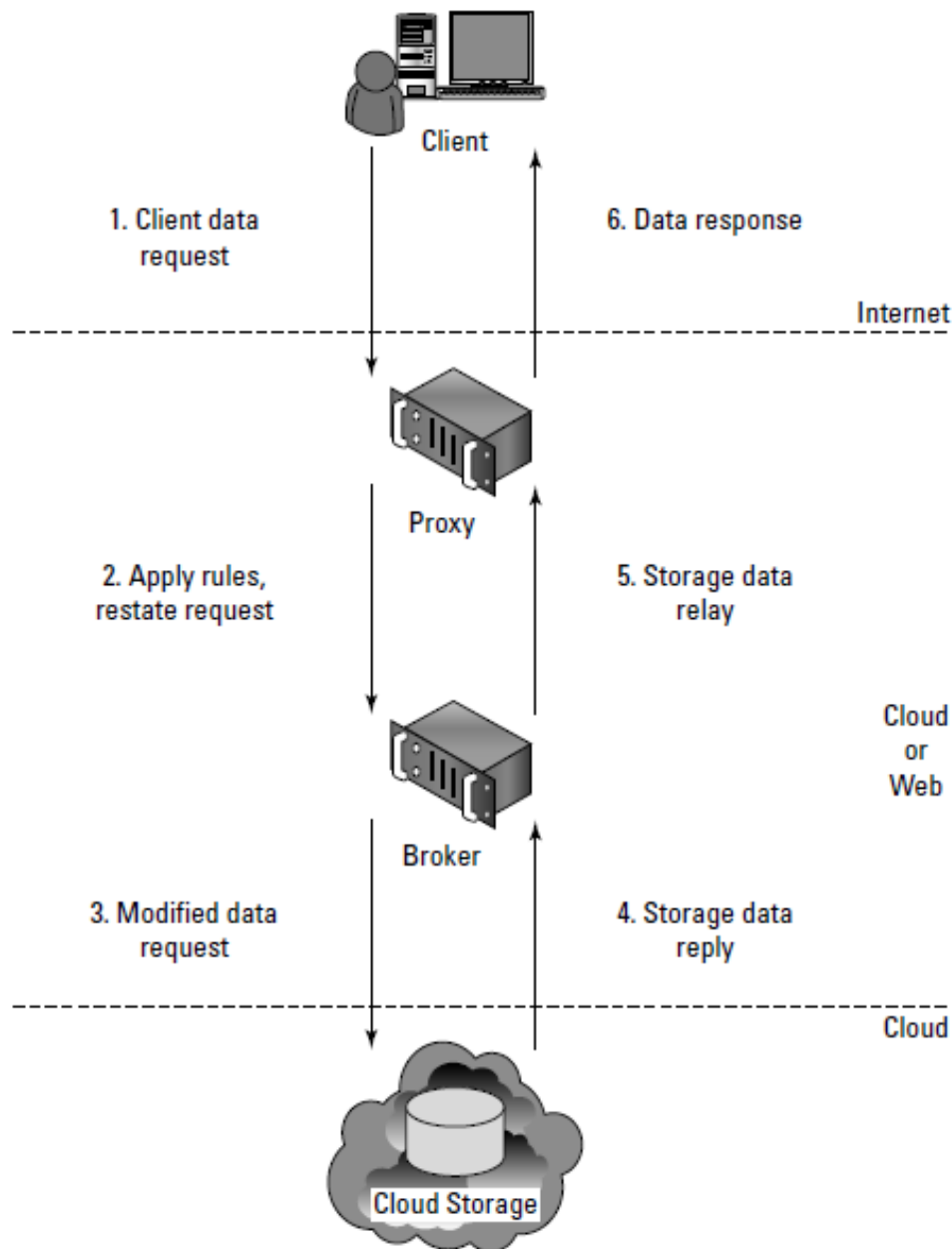
installation in a virtual environment of the VMs that the OVA encompasses.” EX1016, [0021]. This early description is consistent with how the OVF standard describes packaging a virtual appliance. *See* EX1019, 9 (“The OVF descriptor is an XML document that describes meta-data about the software installed on the virtual disks.”); *id.* (“The OVF environment is an XML document containing deployment time customization information for the guest software.”). EX1003, ¶38.

The Open Virtualization Format (OVF) and Open Virtual Appliances (OVAs) continue to be used in the art for deploying / provisioning virtual appliances. *E.g.*, EX1017, [0001]-[0005] (describing systems and methods for “deploying a virtual processing instance from an OVA template file”); EX1018, [0015] (describing “a virtual appliance that can be seamlessly provisioned” for different platforms and “distributed in a known format, such as the Open Virtualization Format (OVF)”); EX1003, ¶39.

3. Midservers Were Well-Known And Facilitated Data Transfer Between Local Networks And Cloud Platforms

It was well-known to use an intermediate system—whether termed a “midserver,” “edge proxy,” “cloud agent,” or “cloud gateway”—as an interface to facilitate data transfer from an enterprise network to a cloud system. *E.g.*, EX1001, 11:28-32 (describing “midserver” as “a gateway to the cloud-based service”); EX1004, [0048] (describing “on-premise cloud agent” as a “gateway that collects data items” and “move[s] the on-premise data to a cloud platform”); EX1007, 258

(“proxy”); EX1012, [0050] (“edge proxies”); EX1013, [0127] (“cloud gateways”); EX1015 (Antony), 6:15-19 (describing connection between “on-premise gateway” and “cloud-side gateway”); EX1022, [0016] (“cloud gateway”). POSITAs understood that using such an interface would simplify the connection between a local network and a cloud network (e.g., providing a single connection rather than individual connections to every device on the network), while ensuring security and other network services. EX1003, ¶40. For example, Sosinsky describes and shows a proxy, which “can be local or in the cloud,” “in the direct data path between the client and data stored in the cloud” (EX1007, 257), as shown in its Figure 12.4:



EX1007, 258; *see also id.*, 257 (describing example where proxy referred to as a “Gatekeeper”). EX1003, ¶40.

Such interfaces were commonly implemented, like other aspects of cloud systems, as a virtual appliance. For example, a 2017 published patent application described “two edge proxies connecting a remote ‘cloud’ network to a local network

that is typically on the premises of a particular entity and termed an ‘on-prem’ network,” where an “*on-prem edge proxy is a virtual appliance* that resides on premises (on-prem),” and “controls network traffic to and from a cloud edge proxy.” EX1012 (Ben-Shaul), [0050], [0071] (emphasis added).

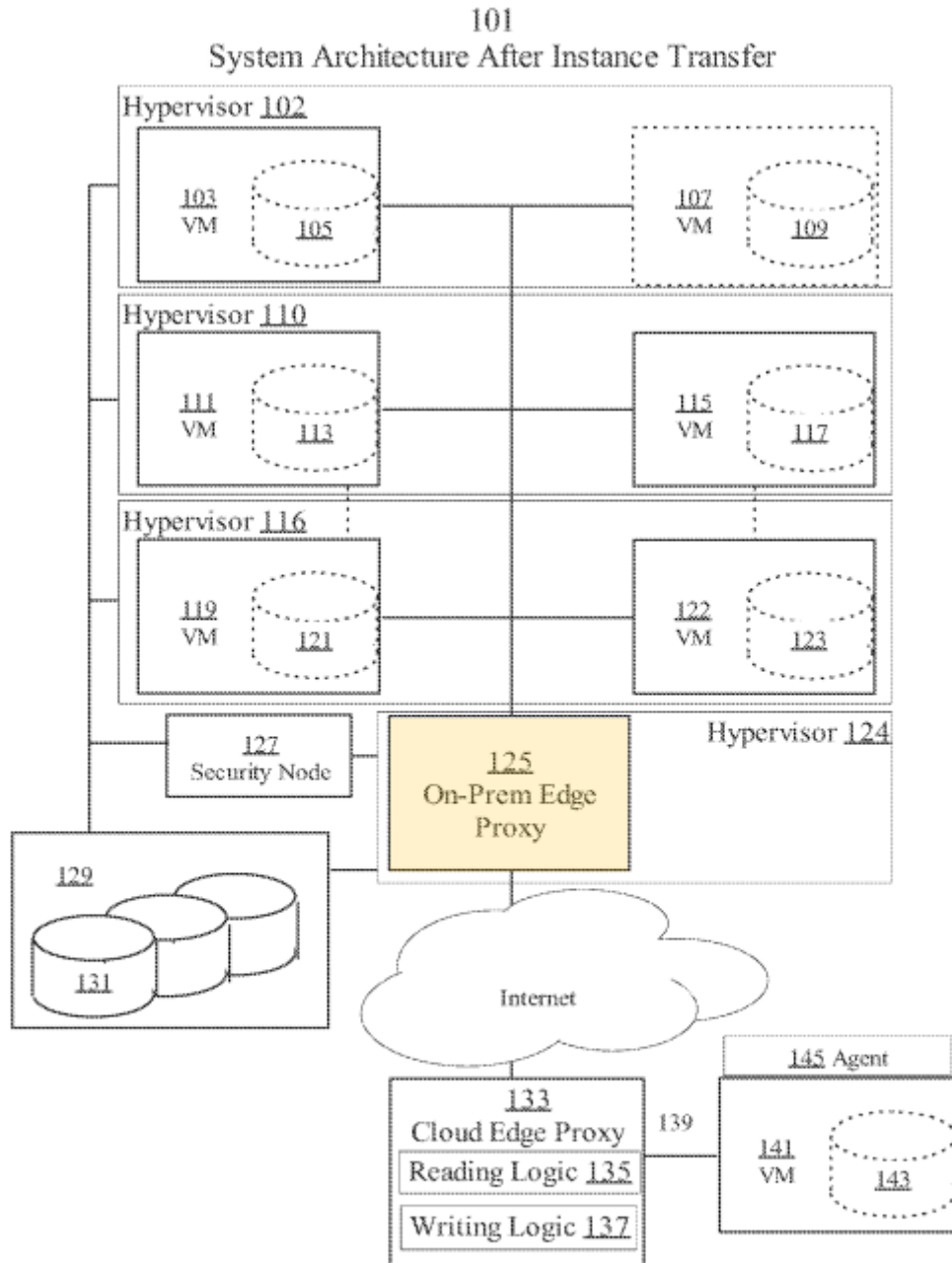


Fig. 1

EX1012, FIG. 1 (highlighting added); EX1003, ¶41.

Similarly, others described a “first cloud gateway in a private cloud” and a “second cloud gateway in a public cloud” connected with a secure transport layer

tunnel over a public network to enable secure network traffic between the private cloud and the public cloud (together forming a “hybrid cloud”). EX1022 (Chang), [0028]; *id.*, [0002] (describing hybrid clouds), [0019]-[0020] (same), [0033]; *see also* EX1015 (Antony), 4:8-20 (describing a gateway “executing as a virtual appliance” that provides components “with connectivity to an external network”); EX1003, ¶42.

B. Person Of Ordinary Skill In The Art

The person of ordinary skill in the art in May 2019 (“POSITA”) would have had a bachelor’s degree in computer science, computer engineering, electrical engineering, or an equivalent degree, with two to four years’ experience designing, implementing, or otherwise working with cloud computing systems. More education, e.g., a Master’s or Ph.D., would compensate for less work experience and vice versa. EX1003, ¶¶27-28.

A POSITA would have been familiar with common, then-existing technologies for cloud computing, including virtualization, as well as issues surrounding the design, implementation, and maintenance of such technologies. The state of the art, above, identifies certain technologies within the knowledge and skill of a POSITA. EX1003, ¶29.

V. THE '663 PATENT

The '663 patent issued December 27, 2022, from an application filed May 14, 2019. It alleges priority to a string of continuation-in-part applications that date back to April 28, 2016, and two 2017 provisional applications.

A. The Claims' Earliest Possible Priority Date Is May 14, 2019

Although the '663 patent alleges priority to forty-three earlier applications (EX1001, 1-2), the May 14, 2019, application was the first to include the subject matter claimed by the '663 patent. The Examiner confirmed that this subject matter was not in any earlier "related" filing. EX1002, 364-65/497, 420-21/497. As a specific example, the earlier applications lack any disclosure of the claimed virtual appliance software application, or virtual appliances generally. In district court, Patent Owner alleges a priority date of May 14, 2019, further confirming that the '663 patent is not entitled to an earlier priority date. EX1023, 9. EX1003, ¶43.

B. The '663 Patent's Specification

The '663 patent describes using "midservers located between the business enterprise computer infrastructure and the cloud-based infrastructure to collect, aggregate, analyze, transform, and securely transmit data from a multitude of computing devices and peripherals at an external network to a cloud-based service." EX1001, Abstract. EX1003, ¶44.

The '663 patent describes “midserver” as “a server that functions as an interface between an external network and a cloud-based service,” and performs standard operations such as “collecting, aggregating, analyzing, filtering, transforming, and securely transmitting data.” EX1001, 12:66-13:5. The “midserver deployment, automated onboarding of data, and endpoint transparent data transport” are described only in terms of conventional applications of known technologies such as Open Virtual Appliance (OVA) and virtual private network (VPN), as well as known commercial and open-source software such as ZeroTier, Ansible, and Nginx. EX1001, 16:11-60; *id.*, 9:51-52 (describing using “standard VPN or similar encryption-based network transport technologies”); EX1003, ¶45. The '663 patent describes the midserver as located “on-site” with the user devices / endpoints from which it collects data (as shown in FIG. 1, reproduced below).

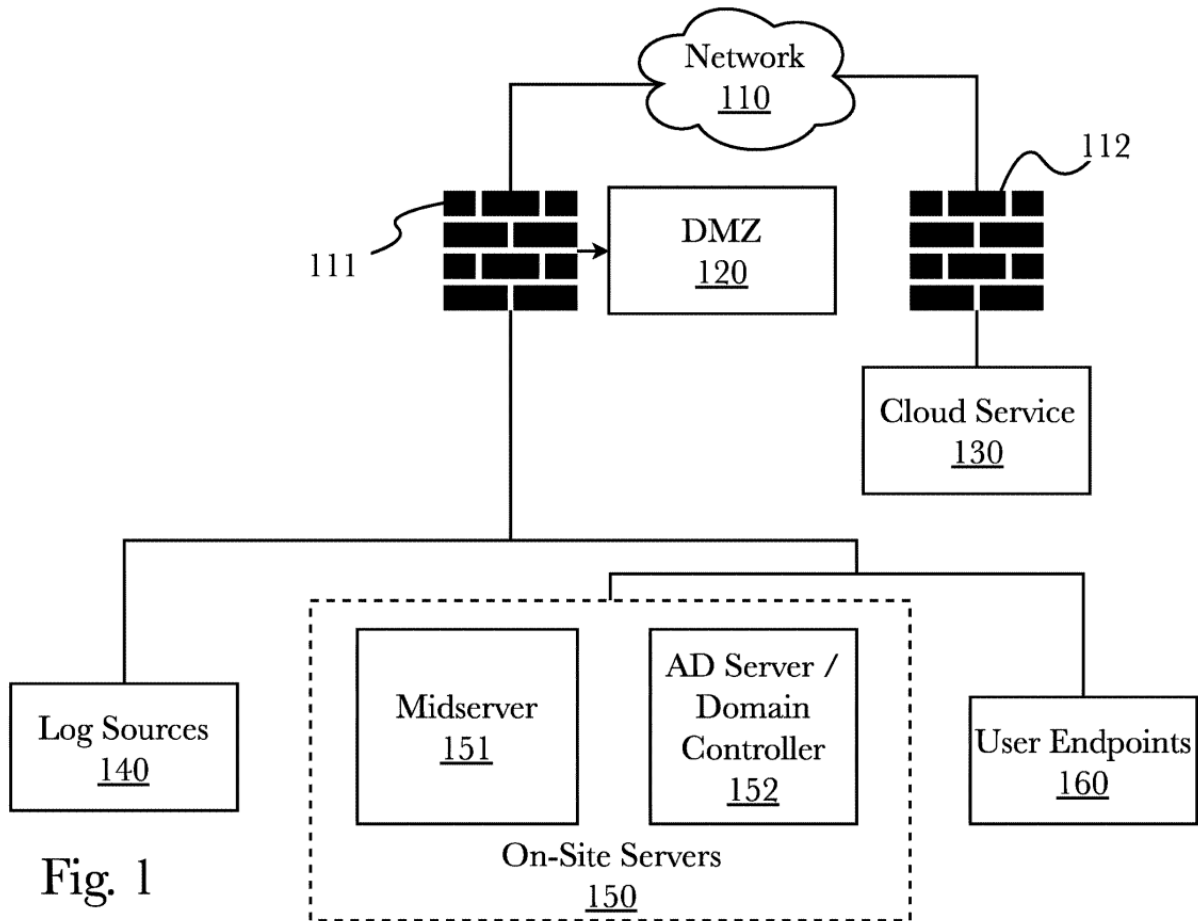


Fig. 1

EX1001, FIG. 1; *id.*, 13:7-43. EX1003, ¶46.

C. The Prosecution History

In a first Office action issued May 4, 2022, the Examiner concluded that the presented claims were not entitled to an earlier filing date because the alleged parent applications did not disclose, *inter alia*, a “midserver.” EX1002, 364-65/497. The Examiner rejected presented claims 1-6, 8, and 9 as anticipated by U.S. 2018/0115519 to Bonomi et al. (“Bonomi”), claim 7 as obvious in view of Bonomi and two non-patent literature references “Himes” and “MatthiasLohr,” and claim 10

as obvious in view of non-patent literature reference “Microsoft” in view of Bonomi.
EX1002, 365-72/497.

In a response filed May 31, 2022, Applicant amended claims 1 and 4 “to more clearly teach the functions of the midserver,” with alleged support in paragraphs [0032]-[0041] and [0060] of the application. EX1002, 415/497. Regarding claim 1, Applicant argued that the newly added features such as a virtual appliance software application and transforming of data (e.g., to encrypt the data) distinguished the claims from the cited prior art Bonomi:

This midserver operations clearly distinguishes over the reference of Bonomi (and the additional references cited below in further rejections), as Bonomi merely teaches a form of firewall that selectively receives and blocks connections between local and external networks (Bonomi, Abstract and at least ¶[0030]). Bonomi **does not provide an automatically-deployed virtual appliance software application that receives *and transforms* multiple data streams for retransmission** over a single, secured data stream to an external host.

EX1002, 415/497 (italics original, bold added). Applicant cancelled claims 7 and 10. EX1002, 416/497.

A second Office action issued June 16, 2022, indicating claims 1-6 as allowable, without comment, and maintaining the rejections of claims 8 and 9

because Applicant did not provide amendments or arguments for those claims. EX1002, 420-23/497. In allowing dependent claims 2, 3, 5, and 6, the Examiner erred by failing to recognize antecedent basis issues caused by the deletions and additions of claim elements in independent claims 1 and 4. *See* EX1002, 411-12/497.

Further, the Examiner's search notes indicate that the Examiner used the exact same search queries before and after the claim amendments, with no apparent change in search strategy to reflect the new elements added to the independent claims. EX1002, 426-30/497.

Applicant cancelled claims 8 and 9 in a response filed July 27, 2022. EX1002, 437-442/497. The Examiner issued a Notice of Allowance on August 10, 2022, without any further comments on the allowability of claims 1-6. EX1002, 444-50/497.

D. The Challenged Claims

Challenged claim 1 is representative and listed here, with element numbering added in the lefthand column:

[1.1]	A system for ingestion of data into a cloud-based service from an external network, comprising:
[1.2]	a midserver comprising at least a processor, a memory, and a plurality of programming instructions stored in the memory and operating on the processor, wherein the plurality of programming instructions, when operating on the processor, cause the processor to:
[1.3]	automatically install a virtual appliance software application, the virtual appliance software application configured to automatically load a plurality of stored configurations on the midserver;

[1.4]	establish a secure network connection to an external network;
[1.5]	receive data over a local network from a plurality of computing devices;
[1.6]	apply a plurality of transformations to at least a portion of the received data; and
[1.7]	retransmit the received data over the secure connection as a single data stream.

A list of challenged claims is provided in Exhibit 1024.

VI. CLAIM CONSTRUCTION

For *inter partes* review, claim terms should be given the ordinary meaning that the terms would have to a person of ordinary skill in the art on the earliest effective filing date, in view of the specification and file history. 83 Fed. Reg. 51340; *Phillips v. AWH Corp.*, 415 F.3d 1303, 1312-1313 (Fed. Cir. 2005) (*en banc*).

Unless otherwise expressly discussed, Petitioner applies the plain and ordinary meaning of all claim terms below. Petitioner does not, however, waive any argument in any litigation that claim terms in the '663 patent are indefinite or otherwise invalid, nor does Petitioner waive its right to raise additional issues of claim construction in any litigation. Petitioner will apprise the Board of any briefing or orders that reflect such positions. EX1003, ¶47.

VII. GROUND 1: CLAIMS 1 AND 4 ARE OBVIOUS OVER MATURANA IN VIEW OF KASHYAP

As explained below, claims 1 and 4 are obvious over Maturana in view of Kashyap. EX1003, ¶48.

A. Maturana (EX1004)

Maturana is a U.S. patent application publication that published October 1, 2015, from an application filed October 27, 2014, and alleges priority to a provisional application filed on March 26, 2014. EX1004, Cover. EX1003, ¶49. Maturana qualifies as prior art under at least Sections 102(a)(1-2).

Maturana discloses “an on-premise cloud agent that facilitates collection and upload of industrial data to a cloud-based storage and processing infrastructure.” EX1004, [0002]. As shown in FIG. 1, Maturana contemplates such cloud agents installed and running at several locations, with each cloud agent collecting data from many different devices:

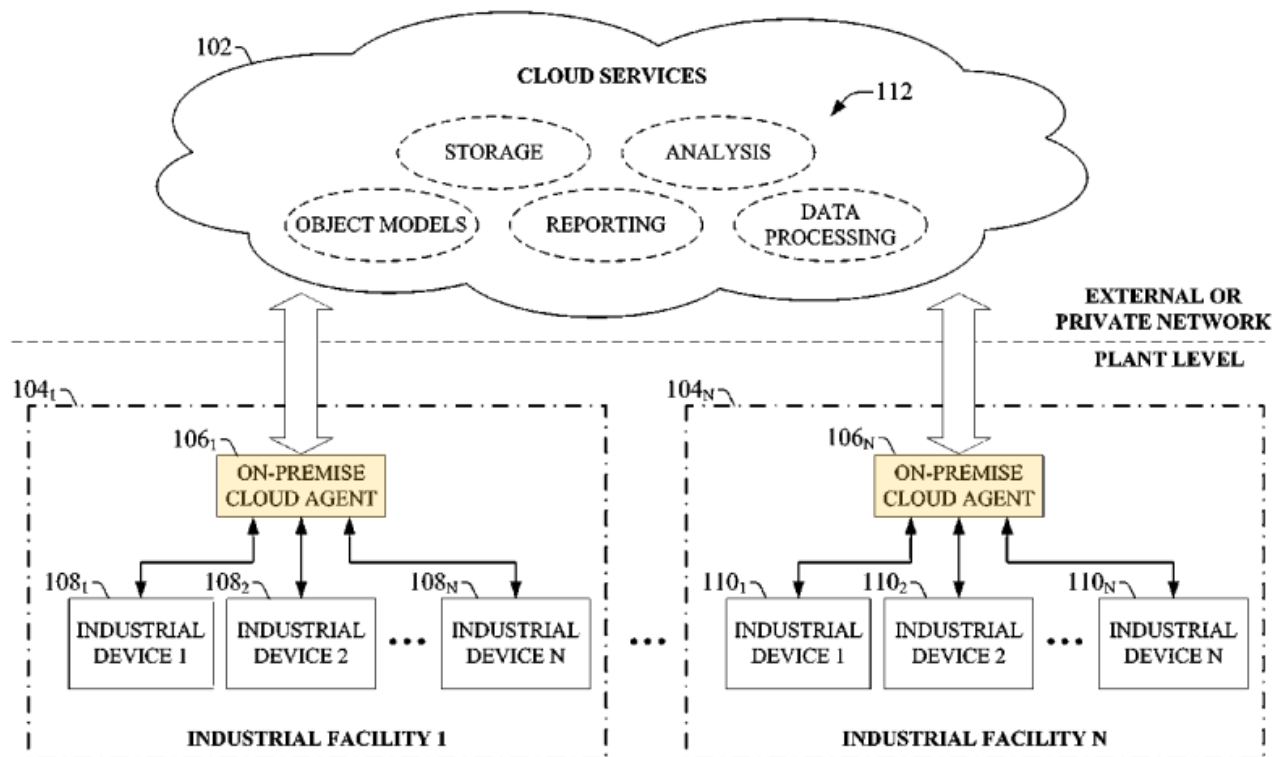


FIG. 1

EX1004, FIG. 1 (highlighting added). The “agent-based cloud infrastructure” of Maturana provides “data ingestion into the cloud.” EX1004, [0046]. EX1003, ¶50.

Maturana explains that “[i]ngestion of industrial device data in the cloud platform 102 through the use of cloud agents 106 can offer a number of advantages particular to industrial automation.” EX1004, [0041]. These advantages include, as some of the examples provided by Maturana, (1) the easy scalability of cloud-based storage “to accommodate the large quantities of data generated daily by an industrial enterprise,” (2) the ability for “multiple industrial facilities at different geographical locations [to] migrate their respective automation data to the cloud ... without the need to establish a private network between the facilities,” and (3) the ability for cloud agents “to automatically detect and communicate with the cloud platform 102 upon installation at any facility, simplifying integration with existing cloud-based data storage, analysis, or reporting applications used by the enterprise.” EX1004, [0041]. Maturana further explains that the cloud platform enables “software vendors to provide software as a service, removing the burden of software maintenance, upgrading, and backup from their customers.” EX1004, [0041]; *see also* EX1003, ¶51.

Maturana discloses implementing its cloud agents using “virtual machine(s).” EX1004, [0042]. The cloud agent of Maturana, as shown in FIG. 2 (reproduced below), includes various components such as a collection services component, a

queue processing component, a subscription management component, a user interface component, one or more processors, and memory. EX1004, [0042]-[0045]; *see also id.*, [0026] (explaining that the components may be software and giving “an application running on a server” as an example of a component).

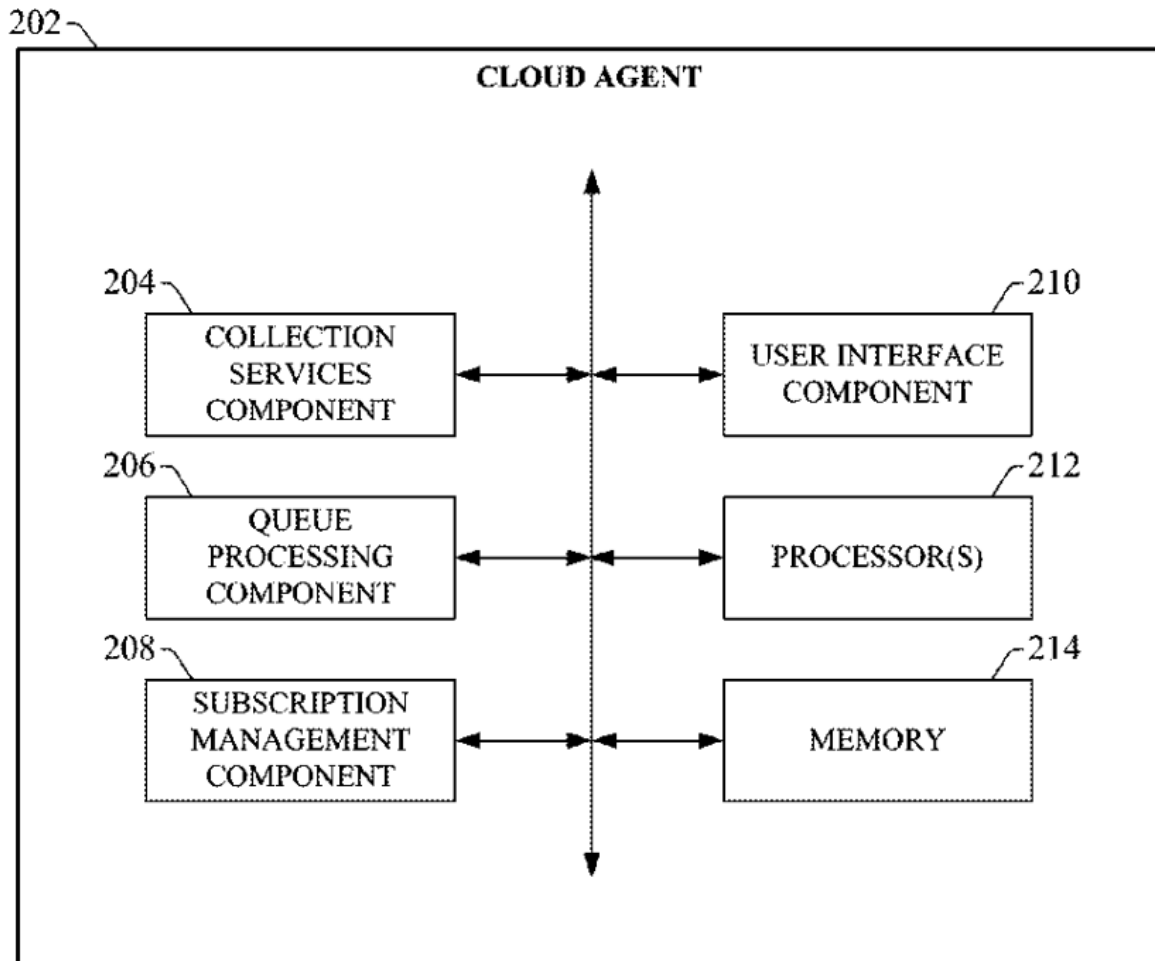


FIG. 2

EX1004, FIG. 2. EX1003, ¶52.

The collection services component is configured to “collect industrial device data, either from a data concentrator or directly from the devices themselves.”

EX1004, [0044]. The queue processing component is configured to “package a compressed data file prepared by the collection services component 204 into a data packet and push the data packet to the cloud platform.” EX1004, [0044]. The subscription management component is “configured to maintain customer-specific configuration and subscription information,” which is used “to determine how the compressed data file should be packaged, and how to connect to the customer’s cloud platform for transmission of the data packets.” EX1004, [0044]; EX1003, ¶53.

B. Kashyap (EX1005)

Kashyap is a U.S. patent application publication that published September 27, 2012, from an application filed March 24, 2011. EX1005, Cover. Kashyap qualifies as prior art under at least Sections 102(a)(1-2). EX1003, ¶54.

Kashyap relates to “configuration of virtual appliances in a virtual machine.” EX1005, [0001]; *see also id.*, [0004]. It describes a “packaging and distribution format” that supports installation and configuration of virtual appliances on virtual machines. *Id.*, [0005], Abstract; *see also id.*, [0021]. Kashyap explains, as was known in the art, that “[a] virtual appliance is a virtual machine image file that includes a pre-configured operating system and a single application.” EX1005, [0003] (“[T]he virtual appliance is a software image containing an application designed to run inside a virtual machine.”). It further teaches that virtual appliances allow users to carry out specific functions on a virtual machine: “A software virtual

appliance is one or more virtual machines that collectively provide a specific function for a user through the services of a virtualization platform.” *Id.*, [0021]. EX1003, ¶55.

Kashyap also describes the mechanics of installing a virtual appliance on a virtual machine: “Deploying a new virtual appliance requires copying the appliance image, including the operating system and application, to a pre-installed and pre-configured virtual machine.” EX1005, [0003]; EX1003, ¶56. It further teaches that “[a] common file format for the virtual appliance is the open virtualization format (OVF).” EX1005, [0003]. It also notes known benefits of OVF, including that it is “platform independent,” allows users to deploy “on a virtualization platform of their choice,” provides a standardized “description and format of virtual image appliances,” and provides features that “allow for additional configuration parameters specific to the particular product.” *Id.*, [0021]. EX1003, ¶57.

Kashyap further teaches a virtual appliance packaging and distribution format that enables efficient “post-install configuration of virtual appliances in a virtual machine.” EX1005, Abstract; *id.*, [0001], [0005]. This format involves installation of both an activation script, for “configuration of the virtual machine on boot” and an associated configuration script, for “configuration of an appliance application.” *Id.*, [0020]. EX1003, ¶58.

Kashyap FIG. 3 (reproduced below) illustrates various aspects of a system that supports “post-install configuration of one or more virtual appliances with use of OVF.” EX1005, [0032] (noting the use of “open virtualization format (OVF)” as a “packaging and distribution format”); *see also id.*, [0012].

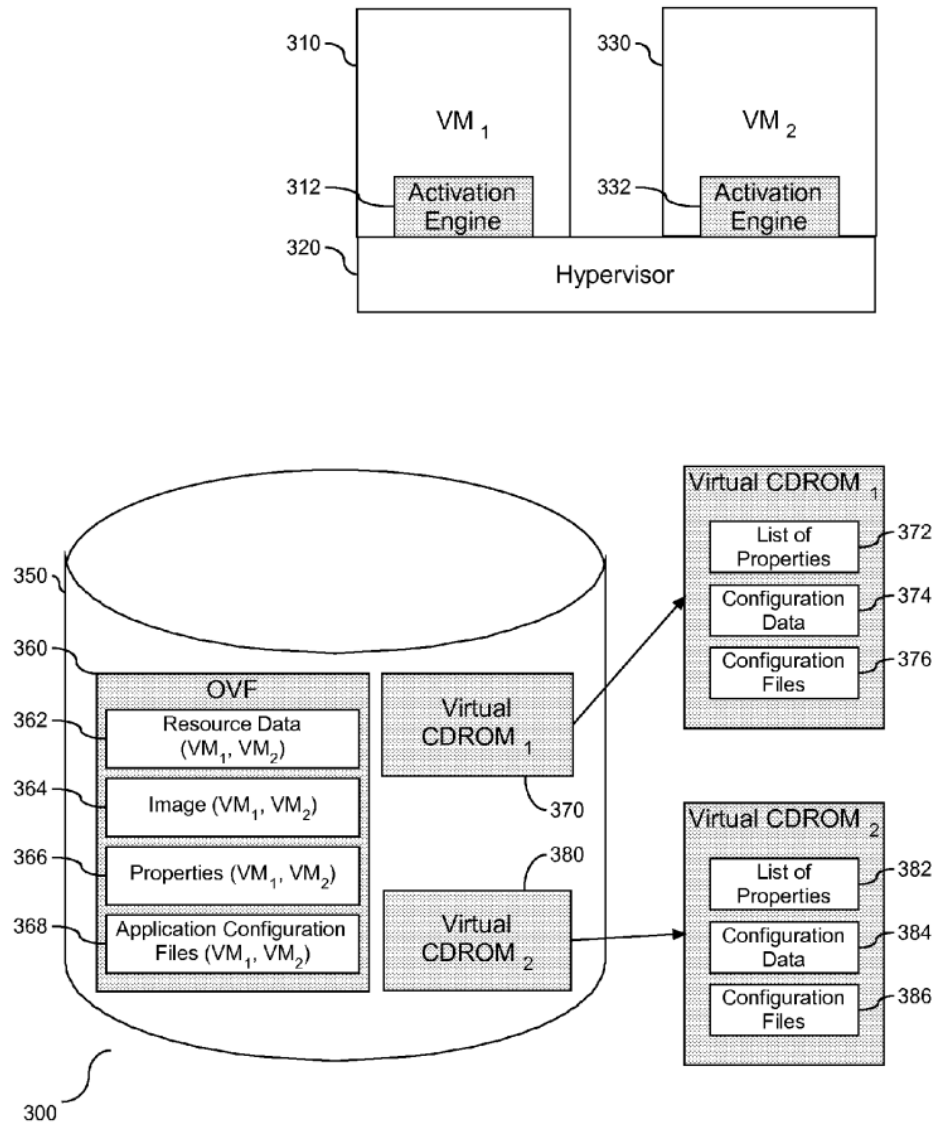


FIG. 3

EX1005, FIG. 3. EX1003, ¶59.

C. Combining Maturana And Kashyap

A POSITA would have been motivated to combine the on-premise cloud agent of Maturana with the automated virtual-appliance deployment and configuration techniques of Kashyap to implement Maturana's cloud agent as a standardized, easily deployable cloud-agent service. The POSITA would have combined these teachings into an improved design (hereinafter "Maturana-Kashyap") that included benefits from each reference. EX1003, ¶60.

So implemented, the Maturana-Kashyap combination includes an agent-based cloud infrastructure with at least the following features:

- A server storing and executing a cloud agent, as taught by Maturana (EX1004, [0048]), configured to:
 - Automatically install the cloud agent as a virtual appliance software application, which automatically loads configuration data while booting, as taught by Kashyap (EX1005, [0003]-[0007], [0020]-[0022]);
 - Securely connect to a cloud platform via an external network, as taught by Maturana (EX1004, [0039], [0057], [0089], FIG. 1);
 - Collect data from a plurality of industrial devices via a local network, as taught by Maturana (EX1004, [0040], [0048]);

- Perform pre-processing and data refinement of the collected data, including aggregating data from multiple sources into a single cloud upload packet, as taught by Maturana (EX1004, [0054]-[0056]); and
- Send the data packet to the cloud platform over the secure connection, as taught by Maturana (EX1004, [0076]).

EX1003, ¶61.

A POSITA would have been motivated, with a reasonable expectation of success, to combine the teachings of Maturana and Kashyap for at least the following reasons. EX1003, ¶62.

First, Maturana discloses cloud agents implemented with “virtual machine(s)” where execution of one or more functional components can “cause the machine(s) to perform the operations described.” EX1004, [0042]. Maturana further describes implementing a cloud agent as “an application running on a server.” EX1004, [0026]. Maturana expressly discloses deploying a persistent on-premise cloud agent that interfaces with multiple local data sources and communicates with a cloud platform, but Maturana does not prescribe a specific deployment or configuration mechanism. *E.g., id.*, [0041] (describing how the cloud agent operates and is configured “upon installation at any facility”); *id.*, [0043]. Kashyap addresses this exact problem by teaching how applications are packaged as virtual appliances and

automatically installed and configured upon instantiation in a virtual machine environment. EX1005, [0003] (describing a “virtual appliance”), [0020]-[0023] (describing automated “application installation procedures” within an OVF framework). As taught by Kashyap and known generally by POSITAs, a virtual appliance comprises “an application designed to run inside a virtual machine.” EX1005, [0003] (describing how to deploy a virtual appliance); *supra* Sections IV.B.1 (explaining that virtual appliances were well known in the art for deploying applications), IV.B.2 (explaining that virtual appliances were known for providing gateway/interface functionality). EX1003, ¶63.

Kashyap teaches that OVF was “a packaging and distribution format for a virtual machine” and “standardizes the description and format of virtual image appliances.” EX1005, [0021]. Within this standardized format, it teaches how to install and configure a virtual machine and associated software application using “activation and configuration scripts.” *Id.*, [0020]. Using this flexible approach, “application installation procedures remain unchanged while allowing for changes and customization of the configuration script independent of the activation script.” *Id.*; *id.*, [0005]-[0007], [0022], [0028]-[0029]. This approach allows a base virtual appliance to be deployed where needed and separately configured for specific instances. EX1003, ¶64.

POSITAs would have thus been motivated to implement Kashyap's detailed teachings on a flexible OVF-based framework for "application installation procedures" for the installation and configuration of cloud agents within the Maturana system. These teachings align with Maturana's virtual-machine execution and "installation at any facility," making them particularly applicable for implementation of the Maturana cloud agent. *E.g.*, EX1004, FIG. 1 (showing "cloud agent[s]" at each site), [0075] (describing the cloud agent being "moved to a different computing device," indicating portability of the cloud agent); *see* EX1005, [0003] (describing a virtual appliance as a packaged application ready for deployment); EX1020, 7-8 (describing the industry standard for packaging and distributing virtual appliances); EX1003, ¶65.

A POSITA would have understood that implementing Maturana's cloud agent as a virtual appliance, as taught by Kashyap, would predictably improve reliability, repeatability, and ease of installation across different environments. Such benefits were well-recognized in cloud and enterprise systems prior to the '663 patent's priority date, particularly for intermediary services that must be deployed on-premise but maintained and updated consistently across customer sites. *Supra* Sections IV.A.1 (describing the benefits of virtual appliances), IV.A.3 (describing known technique of on-premise intermediaries implemented as virtual appliances); EX1003, ¶66.

Second, the teachings of Maturana and Kashyap were complementary for the following reasons. The cloud agent of Maturana is “configured to automatically detect and communicate with the cloud platform 102 upon installation at any facility, simplifying integration with existing cloud-based data storage, analysis, or reporting applications used by the enterprise.” EX1004, [0041]-[0042]. This requirement of Maturana is naturally complemented by Kashyap’s teachings on an OVF-based virtual appliance that is easily installed on virtual machines, while allowing for easy configuration of a particular appliance. For example, Kashyap teaches how to complete installation and configuration of the application in a virtual appliance via activation and configuration scripts. EX1005, [0003], [0005]-[0006], [0020]-[0023]. These teachings of Kashyap naturally complement Maturana’s teaching that a cloud agent should “automatically detect and communicate” with the cloud platform “upon installation at any facility,” because they allow an application such as the cloud agent to be deployed to a server with automatic installation and configuration. EX1004, [0041]; *see also* EX1020, 10 (explaining that the standard format for a virtual appliance provides “full support for unattended installation to ease the burden of deployment for users”); EX1003, ¶67.

Further, Maturana teaches “several cloud agents distributed throughout the customer’s industrial enterprise” and recognizes that data volume “increases as additional plant facilities are added.” EX1004, [0032], [0074]. Kashyap recognizes

the need for “post-install configuration” of virtual appliances decoupled from the virtual appliance itself so that deployed virtual appliances can be customized without a proliferation of customized appliances, and provides a known solution where the configuration script and configuration data are separate from the virtual appliance and loaded upon booting the virtual appliance. EX1005, [0001], [0005]-[0007], [0022], [0027]-[0028], [0031]. A POSITA implementing Maturana would naturally leverage the boot-time configuration taught by Kashyap to easily deploy the same base virtual appliance to different locations while “allowing for changes and customization independent of the” virtual appliance to be applied upon booting. EX1005, [0022]; EX1003, ¶68.

Maturana further discloses “customer-specific configuration and subscription information” that controls packaging and transmission, and registry-stored configurations for secure connectivity. EX1004, [0044], [0052]-[0053], [0069]-[0070], [0074]-[0075]. Kashyap provides the mechanism to provide and apply such parameters automatically, where a “virtual transport mechanism external to the” virtual appliance supplies a plurality of properties, configuration data, and configuration files that the activation startup script and configuration script apply to the virtual appliance at boot. EX1005, [0031]; *id.*, [0020]-[0023], [0027], [0034], FIG. 3. The technique for post-install configuration of the virtual appliance taught by Kashyap thus enables the specific configuration data taught by Maturana to be

automatically provided to the cloud agent upon installation, thereby allowing the cloud agent to “automatically detect and communicate with the cloud platform 102 upon installation at any facility.” EX1004, [0041]; *id.*, [0020], [0022] (“Applications included within virtual appliances may require post install configuration.”). The teachings in Maturana and Kashyap therefore complemented each other, as the automatic installation and configuration techniques taught by Kashyap provide implementation details for achieving the express goals of automatic connection and simplified integration taught by Maturana. EX1004, [0041]; EX1003, ¶69.

Third, Kashyap teaches using the “open virtualization format (OVF)” for packaging and deploying virtual appliances and describes how to implement its post-install configuration technique using OVF. EX1005, [0003], [0021]-[0022], [0026]-[0029], [0032]-[0034]. A POSITA would have been motivated to use the OVF-based virtual appliance technique taught by Kashyap to implement the cloud agent of Maturana because OVF was a widely adopted international standard for packaging and distributing “virtual appliances and the metadata required to *automatically and securely install, configure, and run them* on any virtualization platform.” EX1020, 8 (emphasis added); *id.*, 7; EX1003, ¶70.

Fourth, the teachings of Maturana and Kashyap were compatible. The cloud agent of Maturana is expressly executable on “virtual machine(s)” (EX1004, [0042]) and operate as an “application running on a server” (EX1004, [0026]). Kashyap

expressly describes deploying a virtual appliance onto a virtual machine and completing installation and configuration at boot. EX1005, [0003], [0005]-[0007]. The execution and deployment of a software application is consistent between both Maturana and Kashyap. EX1003, ¶71.

Maturana contemplates that a cloud agent may be “moved to a different computing device,” with registry checks denying communication if the cloud agent is unregistered on the different device. EX1004, [0075]. Maturana thus teaches enforcing cloud agent identity per deployment to ensure secure communication. EX1004, [0074]-[0076]. Kashyap’s teachings, where a virtual appliance is deployed and then specifically configured, are naturally compatible with the agent registry technique of Maturana because the same virtual appliance can be deployed across different locations, while per-site configuration data is loaded at boot as taught by Kashyap. EX1005, [0020]-[0023]; EX1003, ¶72.

A POSITA implementing Maturana thus would have naturally turned to Kashyap for implementation details on how to deploy and customize the cloud agent for a specific site, with the additional advantage of not requiring storage and maintenance of a customized virtual appliance for every site in the cloud. EX1005, [0020] (explaining that “application installation procedures remain unchanged while allowing for changes and customization of the configuration script”); *see, e.g.*, EX1006, [0009] (describing the impracticality of maintaining a plurality of

customized images); EX1007, 52 (explaining that the file size of a virtual appliance may be at least ten times the size of an application alone). This motivation is further reinforced by industry practice recognizing virtual appliances as standard deployment vehicles for cloud-connected services, as they were understood to enable rapid provisioning, automated configuration, and scalable deployment across enterprise infrastructures. *Supra* Section IV.A.1; EX1007, 45; EX1010, 2, 5; EX1014, 279; EX1003, ¶73.

Finally, a POSITA would have reasonably expected to succeed in implementing the Maturana-Kashyap combination because doing so would only entail implementing the respective complementary and compatible teachings of Maturana and Kashyap, as discussed in the preceding paragraphs. At most, a POSITA implementing Maturana-Kashyap would provide software code to implement the respective teachings of Maturana and Kashyap, which was well within the skill of a POSITA. EX1003, ¶74. *E.g., Keynetik, Inc. v. Samsung Elecs. Co.*, No. 2022-1127, 2023 WL 2003932, at *2 (Fed. Cir. Feb. 15, 2023) (citing *Fonar Corp. v. Gen. Elec. Co.*, 107 F.3d 1543, 1549 (Fed. Cir. 1997)) (“Normally, once the function to be performed by software has been identified, writing code to achieve that function is within the skill of the art.”). Further, Kashyap provides specific teachings on using the “open virtualization format (OVF)” to deploy virtual appliances in accordance with its teachings (EX1005, [0006], [0021]-[0023]), and a

POSITA would have reasonably expected to succeed implementing the cloud agent in this way because OVF was a “standard for packaging such virtual appliances and the metadata required to automatically and securely install, configure, and run them on any virtualization platform.” EX1020, 8; *id.*, 7. Further still, similar functionality has long been known in the art of cloud computing, as described above in the state of the art, and a POSITA would have been comfortable applying known and standard techniques. *Supra* Section IV.A; EX1003, ¶74.

So implemented, the Maturana-Kashyap combination includes a self-configuring virtual appliance-based cloud agent that collects data from a plurality of devices, processes the collected data to prepare it for transmission, and transmits the data over a secure connection to a cloud platform, and thus renders all challenged claims obvious. EX1003, ¶75.

D. Claim 1

1. Element [1.1]²

[1.1] A system for ingestion of data into a cloud-based service from an external network, comprising:

² Reference numbers in the format of [claim#.element#] are added throughout for ease of reference.

To the extent the preamble is limiting, Maturana-Kashyap satisfies it. Maturana discloses “an on-premise agent architecture for collection of industrial data from plant-floor devices and transfer of the data to a cloud repository for processing and storage.” EX1004, [0034]. Maturana expressly describes this collection and transfer of data as “*ingesting industrial data to a cloud platform using a cloud agent architecture.*” EX1004, [0019], [0085] (emphasis added). And, as shown in FIG. 1, Maturana teaches that the ingestion of data into the cloud platform (“cloud services”) is done from a network external to the cloud services:

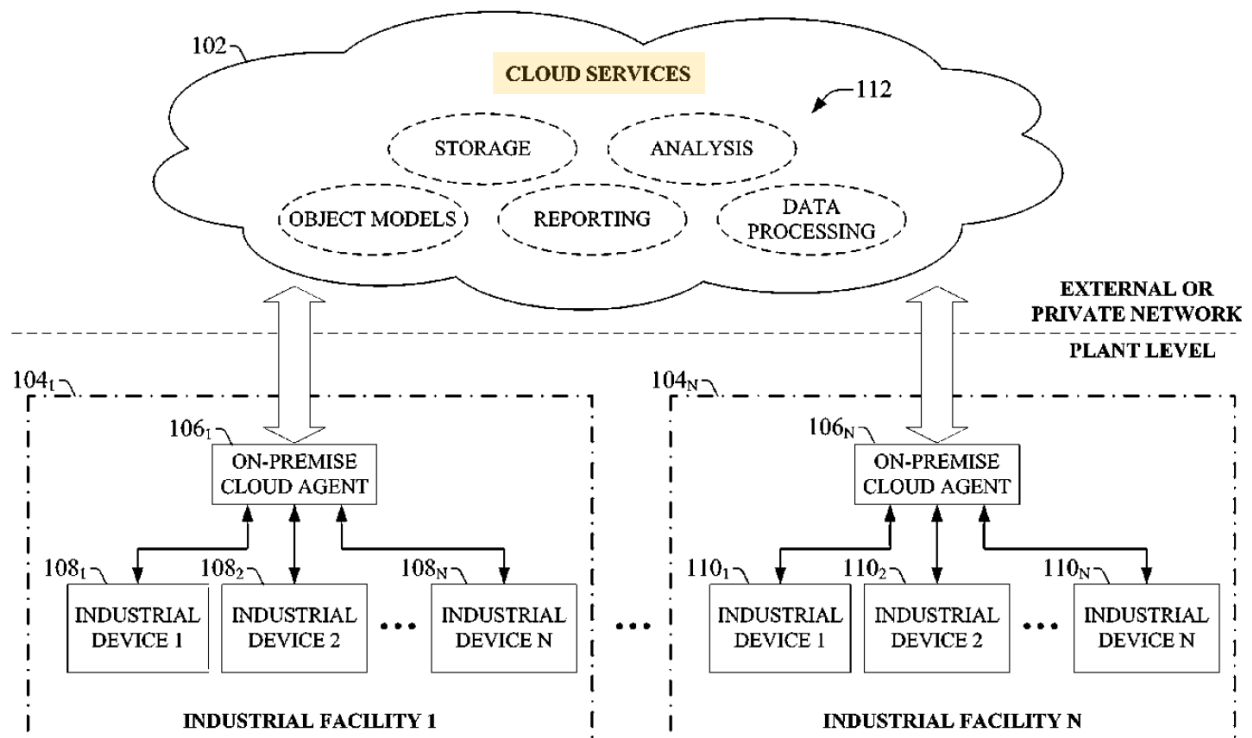


FIG. 1

EX1004, FIG. 1. Maturana labels the cloud services as “external” to the network where the on-premise cloud agents reside and from which data originates. From the

perspective of the cloud, each cloud agent network is an external network. EX1003, ¶76; *see also* EX1004, [0037]; *id.*, [0033]-[0034] (discussing networks on which applications “collect and generate data” and how the cloud agents transfer “the data to a cloud repository for processing and storage”), [0039]-[0041].

2. Element [1.2]

[1.2] a midserver comprising at least a processor, a memory, and a plurality of programming instructions stored in the memory and operating on the processor, wherein the plurality of programming instructions, when operating on the processor, cause the processor to:

Maturana-Kashyap satisfies this element. The server / hardware platform executing the cloud agent taught by Maturana is a *midserver* because it “facilitates collection and upload of industrial data to a cloud-based storage and processing infrastructure” (EX1004, [0002]), just as the midserver of the ’663 patent is used to “collect, aggregate, analyze, transform, and securely transmit data from a multitude of computing devices and peripherals at an external network to a cloud-based service” (EX1001, 7:60-66). *See* EX1004, [0048] (explaining that “[c]loud agent 202 can execute on any suitable hardware platform (e.g., a *server*, a LINUX box, etc.)”). The ’663 patent describes a midserver as “an interface between an external network and a cloud-based service” (EX1001, 8:5-7, 8:17-19), and “[u]sing the midserver as a gateway to the cloud-based service” (EX1001, 11:28-29). The server hosting the cloud agent of Maturana meets this description, as it “acts as a generic

gateway that collects data items from the various industrial assets on the plant network and packages the collected data according to a generic, uniform data packaging schema used to move the on-premise data to a cloud platform 302.”
EX1004, [0048]; EX1003, ¶77.

Maturana depicts the **cloud agent 202** as executing on a hardware **server 808** in FIG. 8:

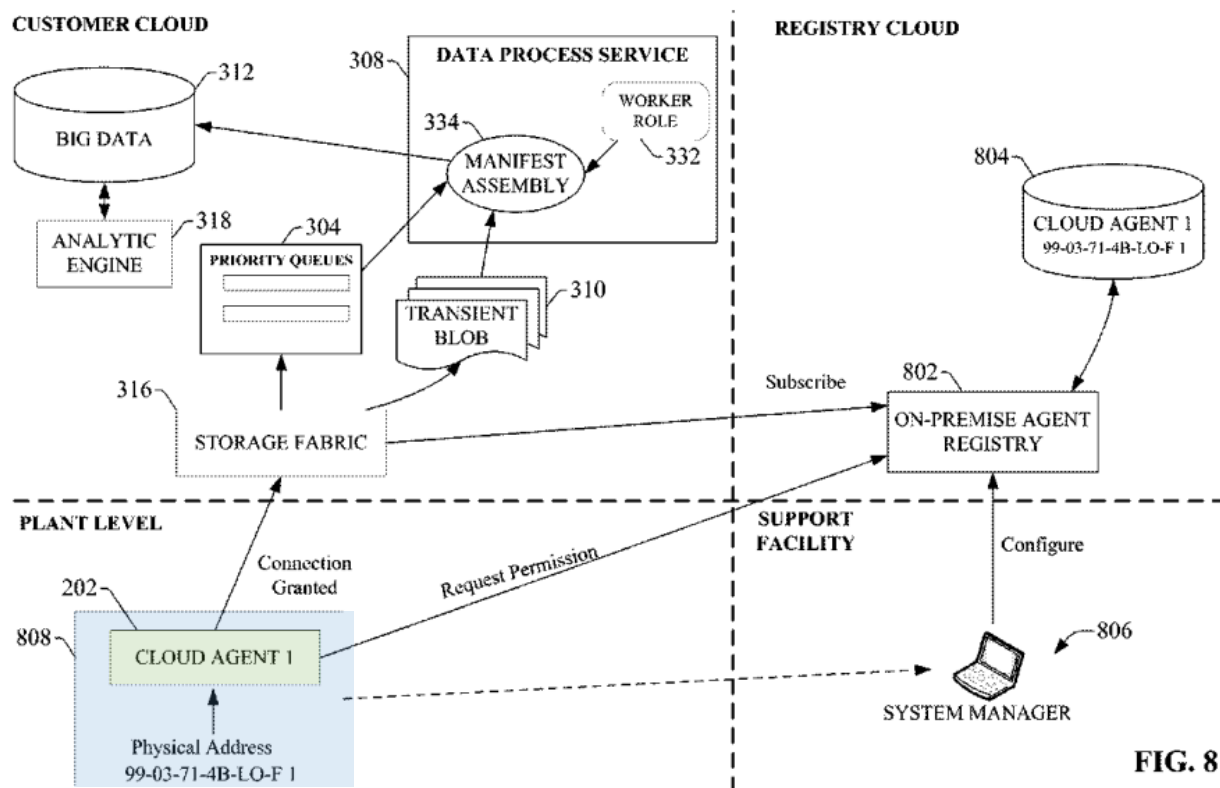


FIG. 8

EX1004, FIG. 8 (highlighting added); *id.*, [0074]; EX1003, ¶78.

Within this hardware host, as depicted in FIG. 2 of Maturana below, the cloud agent 202 includes “one or more **processors 212**, and **memory 214**” (*at least a processor, a memory*), and various functional components that “comprise **software**

instructions stored on memory 214 and executed by processor(s) 212” (a plurality of programming instructions stored in the memory and operating on the processor).

EX1004, [0043] (color added).

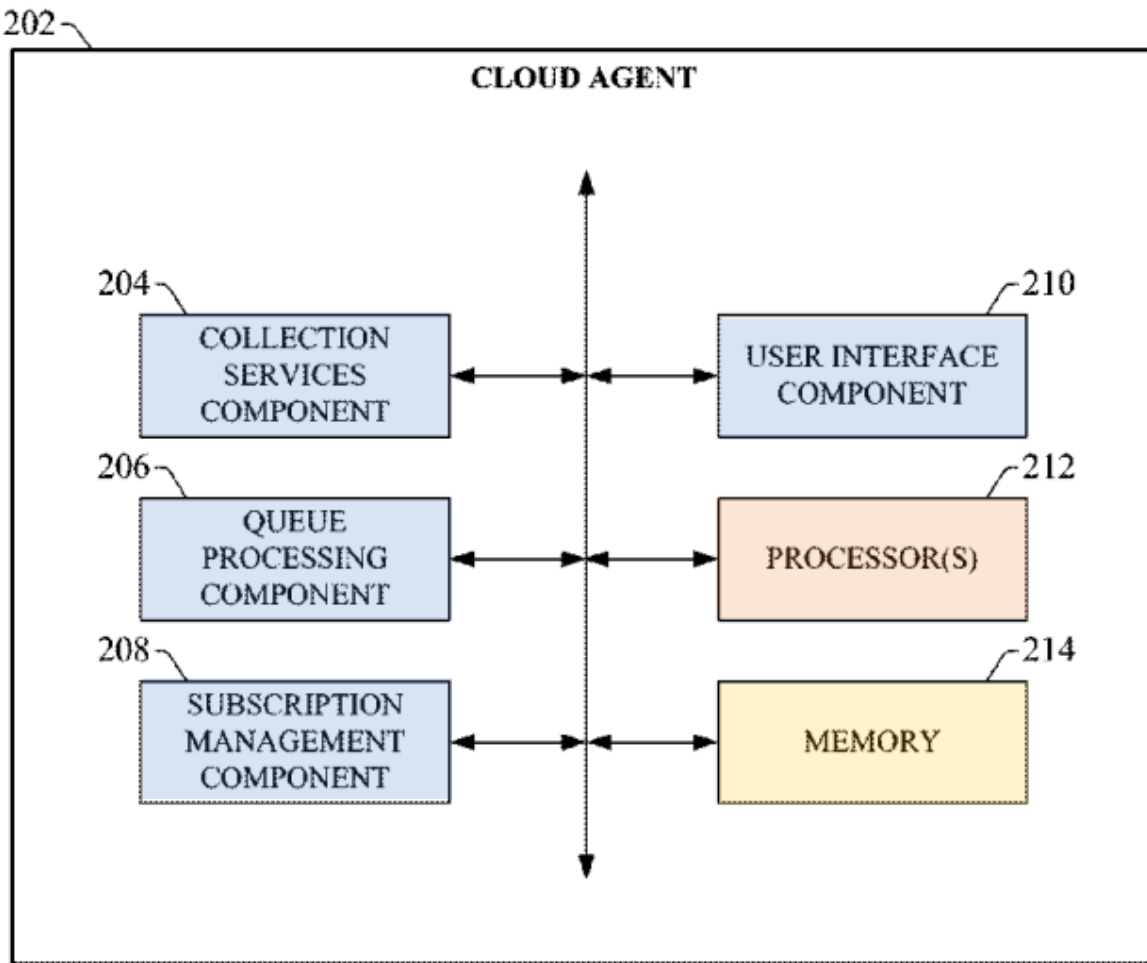


FIG. 2

EX1004, FIG. 2 (highlighting added); EX1003, ¶79.

A POSITA would have understood the underlying gateway computer (e.g., server) with at least one processor and memory as a *midserver*, and the cloud-agent components stored as a virtual appliance in that memory and executed by the

processor to perform the gateway / midserver functions (collection, transformation, and transmission of data to the cloud platform) as *a plurality of programming instructions*. As explained in the following sections, in the Maturana-Kashyap system these instructions cause the midserver to perform the recited functions. EX1003, ¶80.

3. Element [1.3]

[1.3] automatically install a virtual appliance software application, the virtual appliance software application configured to automatically load a plurality of stored configurations on the midserver;

Maturana-Kashyap satisfies this element. As discussed *supra* Section VII.C, a POSITA implementing Maturana's teachings of a cloud agent comprising "an application running on a server" and executed with "virtual machine(s)" (EX1004, [0026], [0042]) would have been motivated to use Kashyap's teachings regarding a standardized way to install and configure "an application designed to run inside a virtual machine," namely via a virtual appliance (EX1005, [0003]). EX1005, [0020]-[0021]. The virtual appliance is deployed to the server of Maturana-Kashyap via OVF, as taught by Kashyap, which involves "copying the appliance image, including the operating system and *the application*, to a pre-installed and pre-configured virtual machine," such as the virtual machine on the server disclosed by Maturana. EX1005, [0003] (emphasis added); *id.*, [0021]; EX1004, [0042]. The cloud agent of Maturana-Kashyap is therefore a *virtual appliance software application* because it

is the software application provided within the virtual appliance, as taught by Kashyap, and executed with a virtual machine, as taught by Maturana. *Supra* Section VII.C; EX1003, ¶81.

Regarding “automatically install a virtual appliance software application,” the server (*midserver*) automatically installs the cloud agent (*virtual appliance software application*) of Maturana-Kashyap when the virtual appliance is deployed to the server. EX1020, 8 (explaining that a virtual appliance distributed via OVF is “automatically and securely install[ed]”); EX1003, ¶82.

Kashyap discloses “application installation procedures” that *automatically install the virtual appliance software application* when the virtual appliance boots. EX1005, [0020]. Specifically, the “application included within a virtual appliance requires installation of activation and configuration scripts” to complete installation of the application. EX1005, [0020]; *id.* [0025]. The activation script, provided with the virtual appliance and “instantiated during boot operation” of the virtual appliance, configures the virtual machine when the virtual appliance boots, while the configuration script, provided separately from the virtual appliance and called by the activation script during boot, configures the application itself. EX1005, [0005]; *id.*, [0006]-[0007], [0020]-[0034]. POSITAs would have understood that these scripts were executed *automatically* because they occur as part of the boot operation. EX1003, ¶81. POSITAs further would have understood the “activation”

step taught by Kashyap as completing the installation of the virtual appliance software application in the specific virtual environment. EX1011, 620; EX1003, ¶83.

This process of activating the virtual appliance is depicted in FIG. 1 of Kashyap:

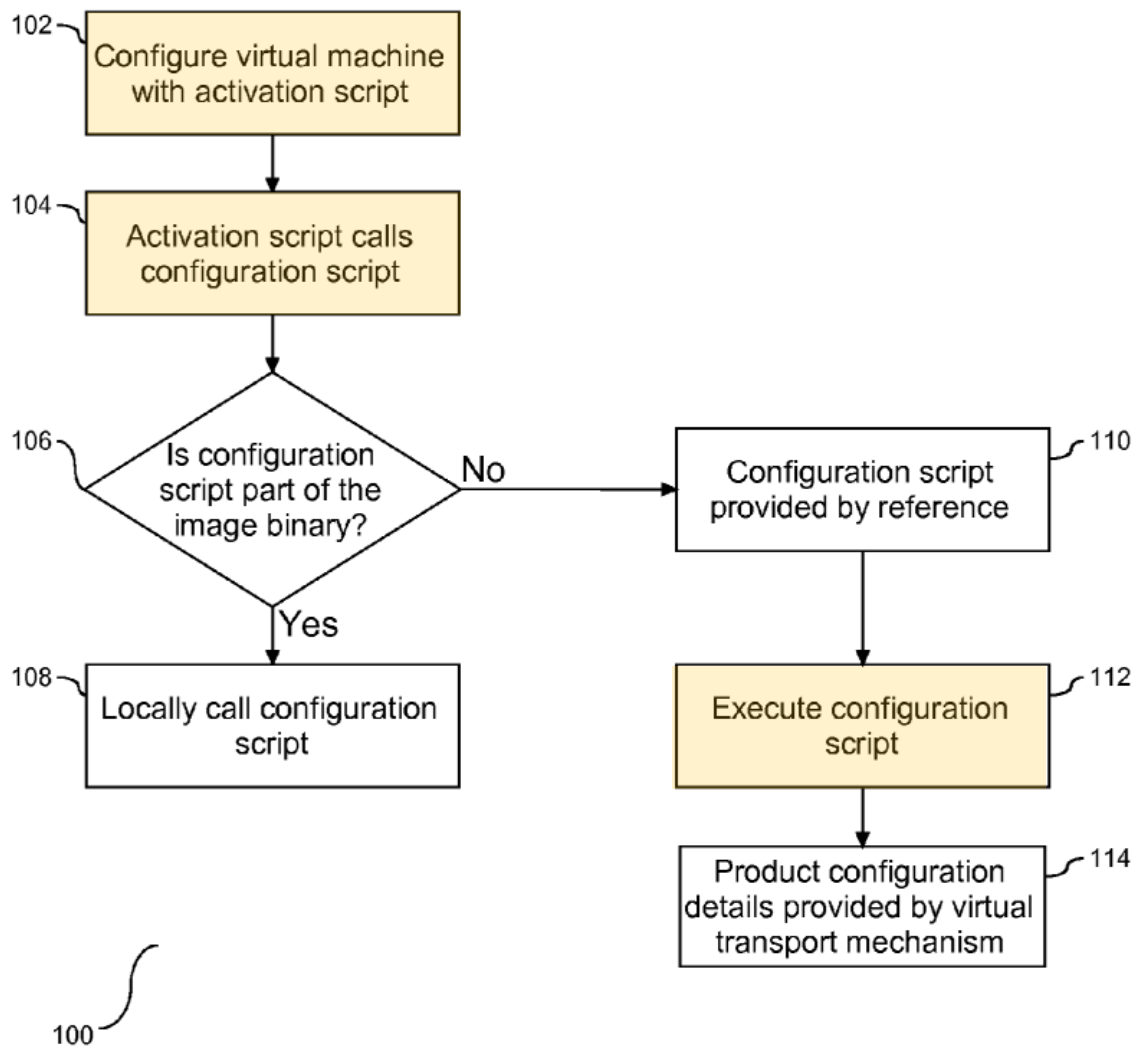


FIG. 1

EX1005, FIG. 1 (highlighting added). Together, these automatically executed scripts *automatically install the virtual appliance software application*. EX1003, ¶84.

Regarding “the virtual appliance software application configured to automatically load a plurality of stored configurations on the midserver,” Kashyap explains that “[a]pplications included within virtual appliances may require post install configuration,” and that “[t]o enable the customization, the configuration framework must not be embedded or hardcoded within the virtual machine binary as this would not support post-install configuration.” EX1005, [0022]. To address this, Kashyap teaches using a virtual transport mechanism specified in an OVF specification to load configuration information into the booting virtual appliance (*the virtual appliance software application configured to automatically load a plurality of stored configurations on the midserver*). EX1005, [0022], [0027]; EX1003, ¶85.

Specifically, during installation of the virtual appliance software application, the activation script calls the configuration script. EX1005, [0020], [0022]-[0023]. The configuration script loads configuration data and configuration files (*a plurality of stored configurations*) via a virtual transport mechanism. EX1005, [0026]-[0028], [0031], [0034]. This configuration data is loaded in one embodiment as a virtual CD-ROM, and used to configure the virtual appliance software application. EX1005, [0005]-[0007], [0020], [0022]. During this process, where the cloud agent is configured using the configuration data provided via the virtual transport

mechanism, the cloud agent (*virtual appliance software application*) thus *automatically loads a plurality of stored configurations on the midserver*. FIG. 1 of Kashyap depicts the process of loading the configuration data:

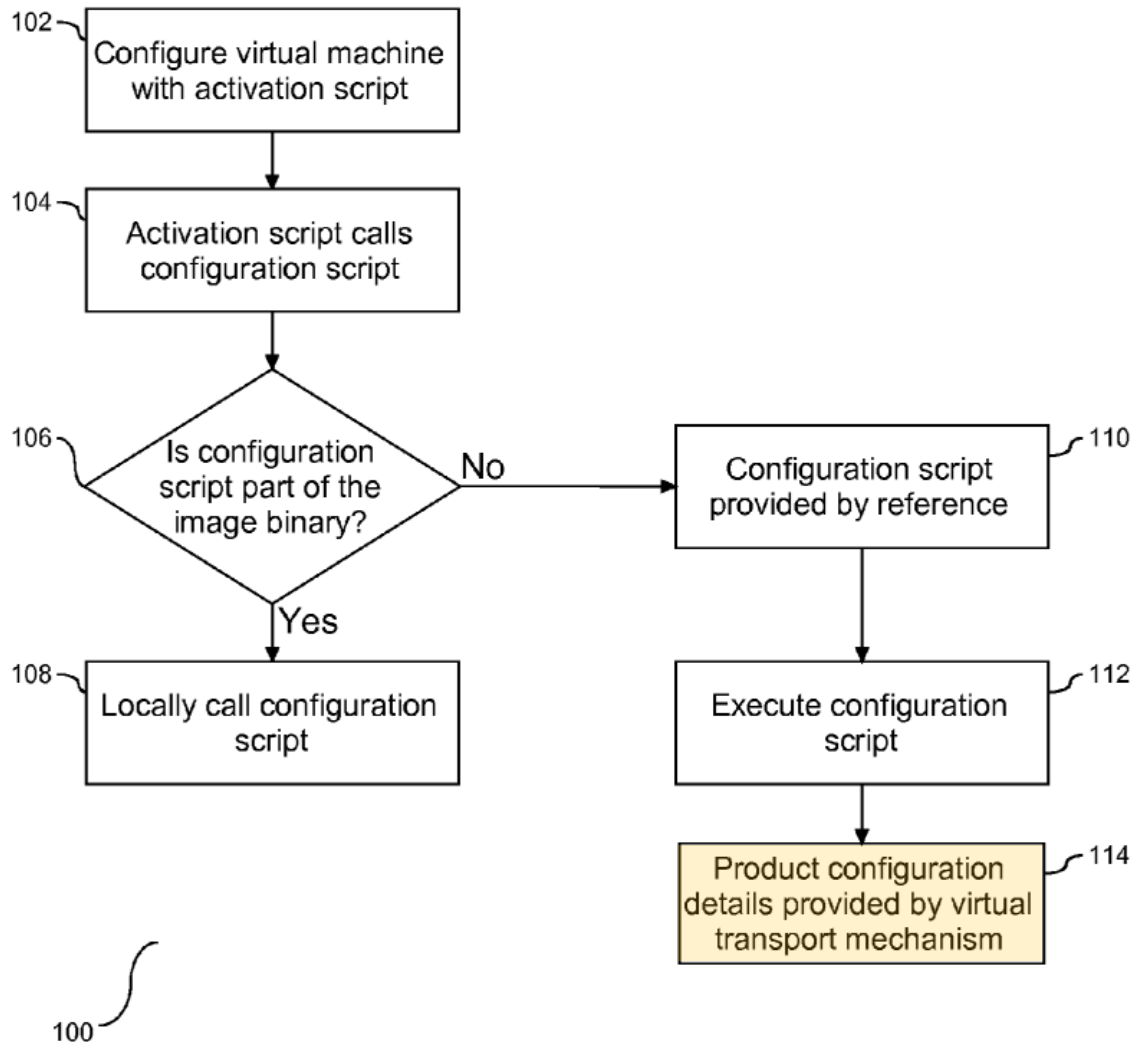


FIG. 1

EX1005, FIG. 1 (highlighting added); EX1003, ¶86.

FIG. 3 of Kashyap depicts a list of properties, configuration data, and configuration files (together, *a plurality of stored configurations*) in a virtual CD-ROM that are used to configure the application:

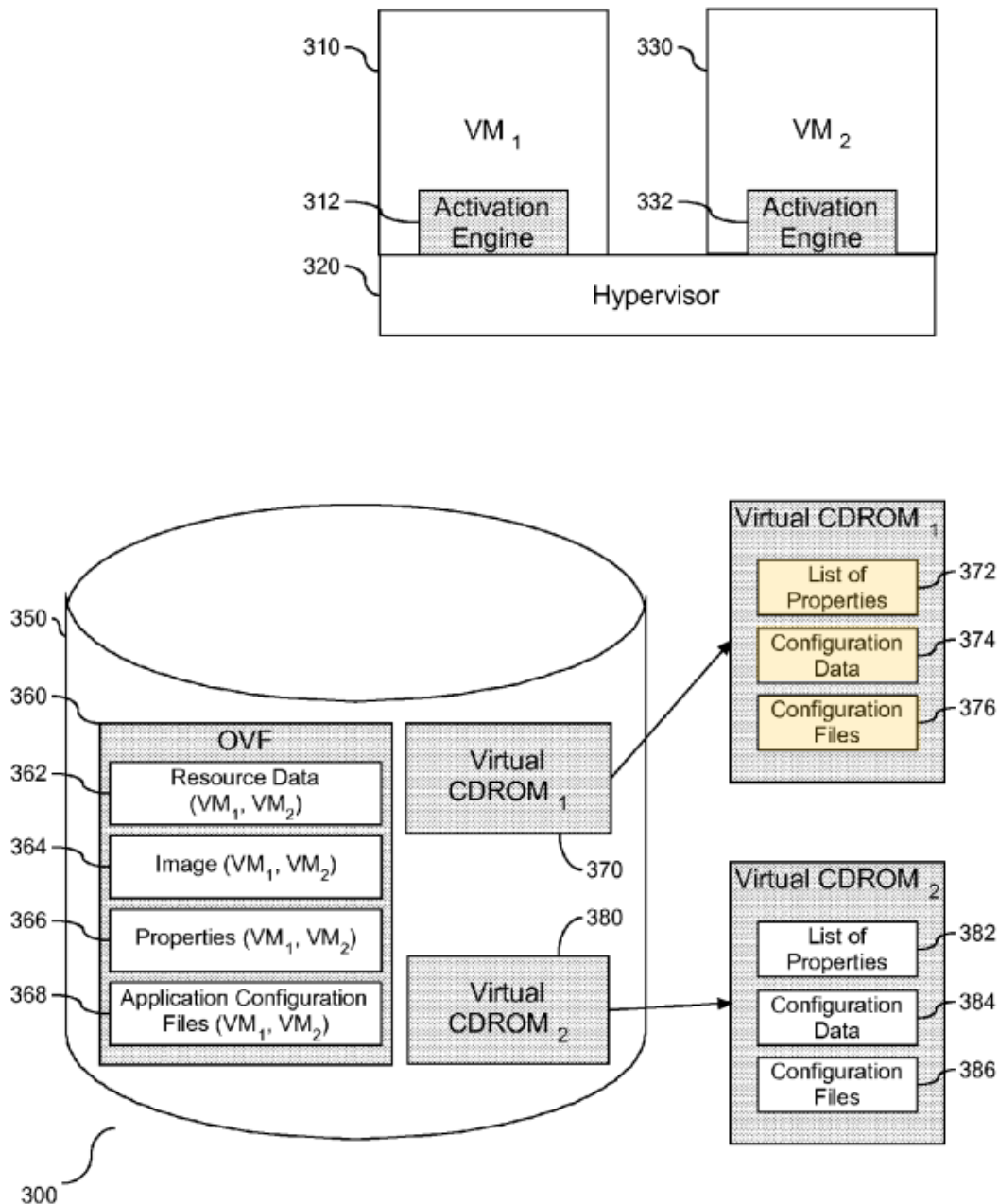


FIG. 3

EX1005, FIG. 3 (highlighting added). In combination, the list of properties, configuration data, and configuration files taught by Kashyap further include “customer-specific configuration and subscription information” taught by Maturana. EX1004, [0044], [0053]; EX1003, ¶87.

So implemented, the cloud agent of Maturana-Kashyap is provided to an on-premise server as a virtual appliance software application that is activated in the virtual machine environment of the server (*automatically install a virtual appliance software application*). The cloud agent (*virtual appliance software application*) then automatically loads configuration data and configuration files, including customer-specific configuration information, as part of its boot process (*the virtual appliance software application configured to automatically load a plurality of stored configurations on the midserver*), where this configuration data enables the cloud agent to “automatically detect and communicate with the cloud platform.” EX1004, [0041]; EX1003, ¶88.

4. Element [1.4]

[1.4] establish a secure network connection to an external network;

Maturana-Kashyap satisfies this element. Maturana discloses establishing a secure network connection to an external network, referred to by Maturana as “the cloud,” through a “methodology for securely connecting a cloud agent to a cloud platform by handshaking with an agent registry” (*establish a secure network*

connection to an external network). EX1004, [0021], [0089]. Specifically, “[t]o ensure *secure outbound traffic to the cloud*, one or more embodiments of cloud agent 202 can support HTTPS/SSL, certificate authority enabled transmission, and/or unique identity using MAC addresses.” EX1004, [0057] (emphasis added); EX1003, ¶89. Maturana depicts the cloud agent connecting to cloud services on an “external network” in FIG. 1:

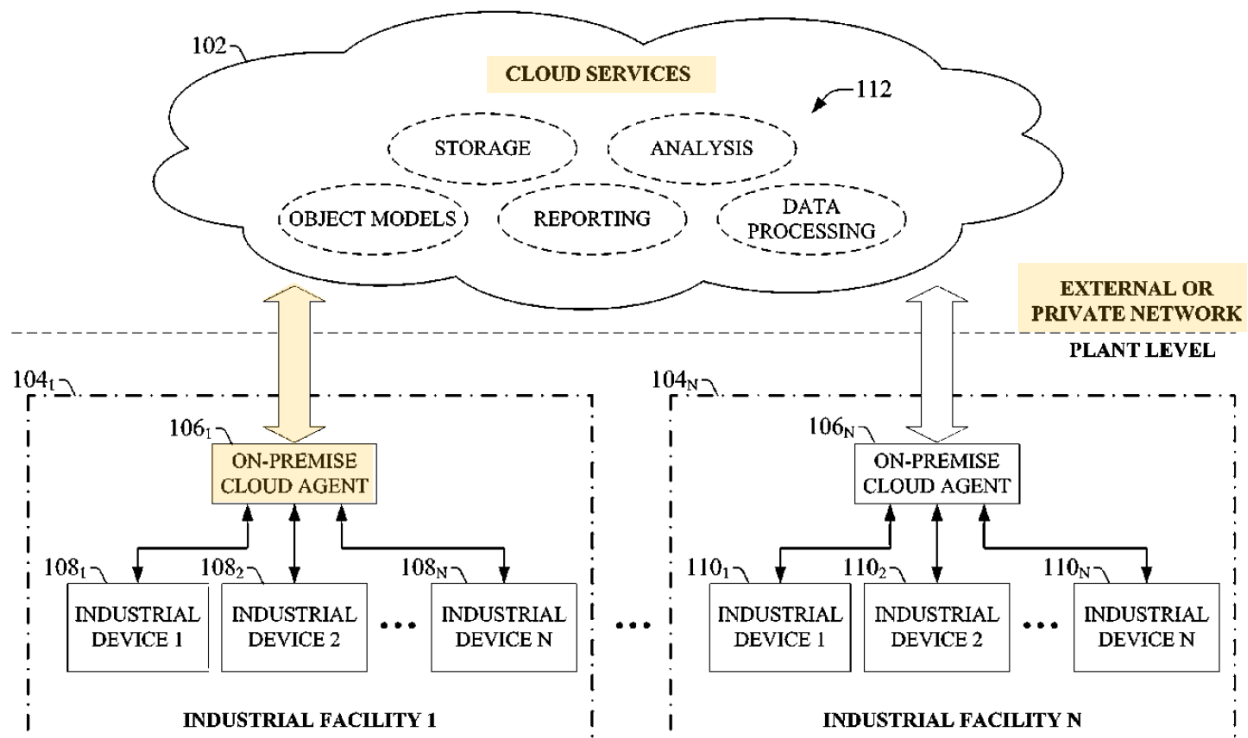


FIG. 1

EX1004, FIG. 1 (highlighting added); EX1003, ¶90.

Maturana-Kashyap thus discloses and teaches establishing a secure network connection to an external network, at least because Maturana expressly discloses doing so. EX1003, ¶91.

5. Element [1.5]

[1.5] receive data over a local network from a plurality of computing devices;

Maturana-Kashyap satisfies this element. Maturana discloses the cloud agent receiving data from a plurality of computing devices over a local network. Specifically, the cloud agent “collect[s] data from industrial devices” (*receive data from a plurality of computing devices*). EX1004, [0039]. Figure 1 of Maturana depicts a plurality of industrial devices providing data to a cloud agent over a local network:

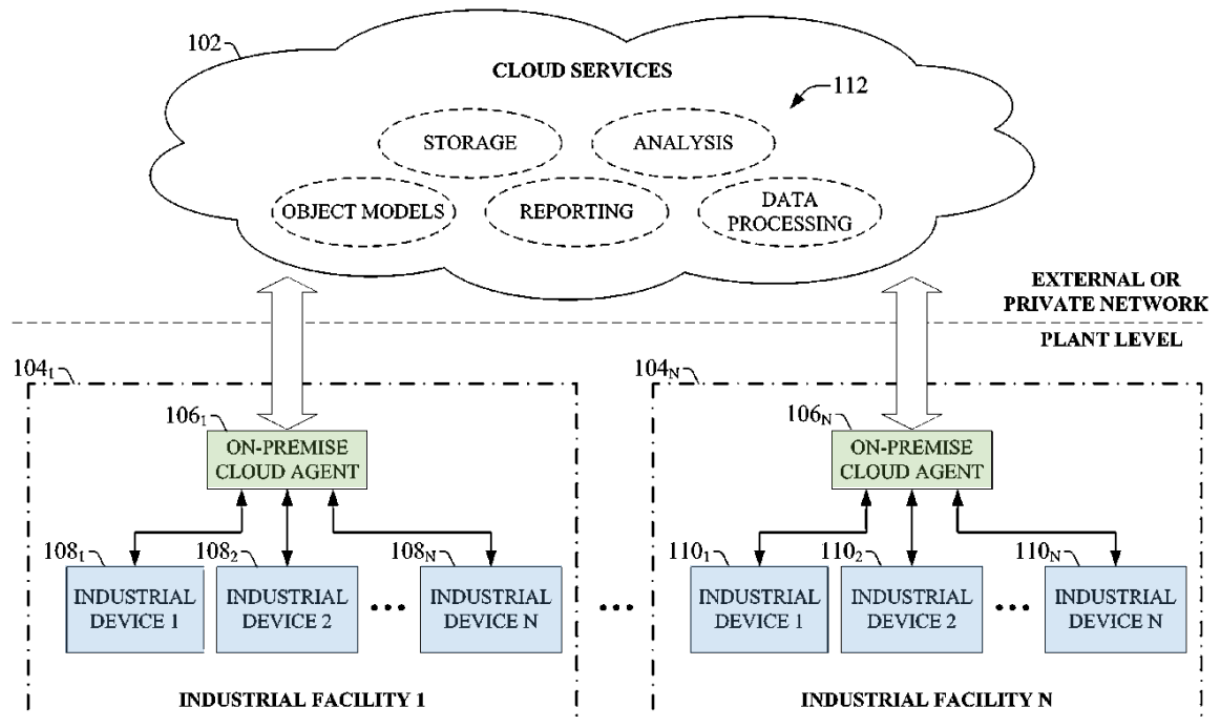


FIG. 1

EX1004, FIG. 1 (highlighting added); EX1003, ¶92.

The “number of industrial devices” (e.g., [industrial device 1](#), [industrial device 2](#), through [industrial device N](#)) providing data to the cloud agent (e.g., [on-premise cloud agent](#)) include “industrial controllers that facilitate monitoring and control of their respective processes” (*a plurality of computing devices*). EX1004, [0037]; *id.*, [0036]. The industrial controllers are *computing devices* because the term “controller” refers to “a computer-related entity,” including “a computer” (EX1004, [0026]), and such “[i]ndustrial controllers store and execute user-defined control programs to effect decision-making in connection with the controlled process” (EX1004, [0031]). FIG. 12 of Maturana, reproduced below, expressly discloses receiving collected industrial data at a cloud agent from one or more on-premise data sources:

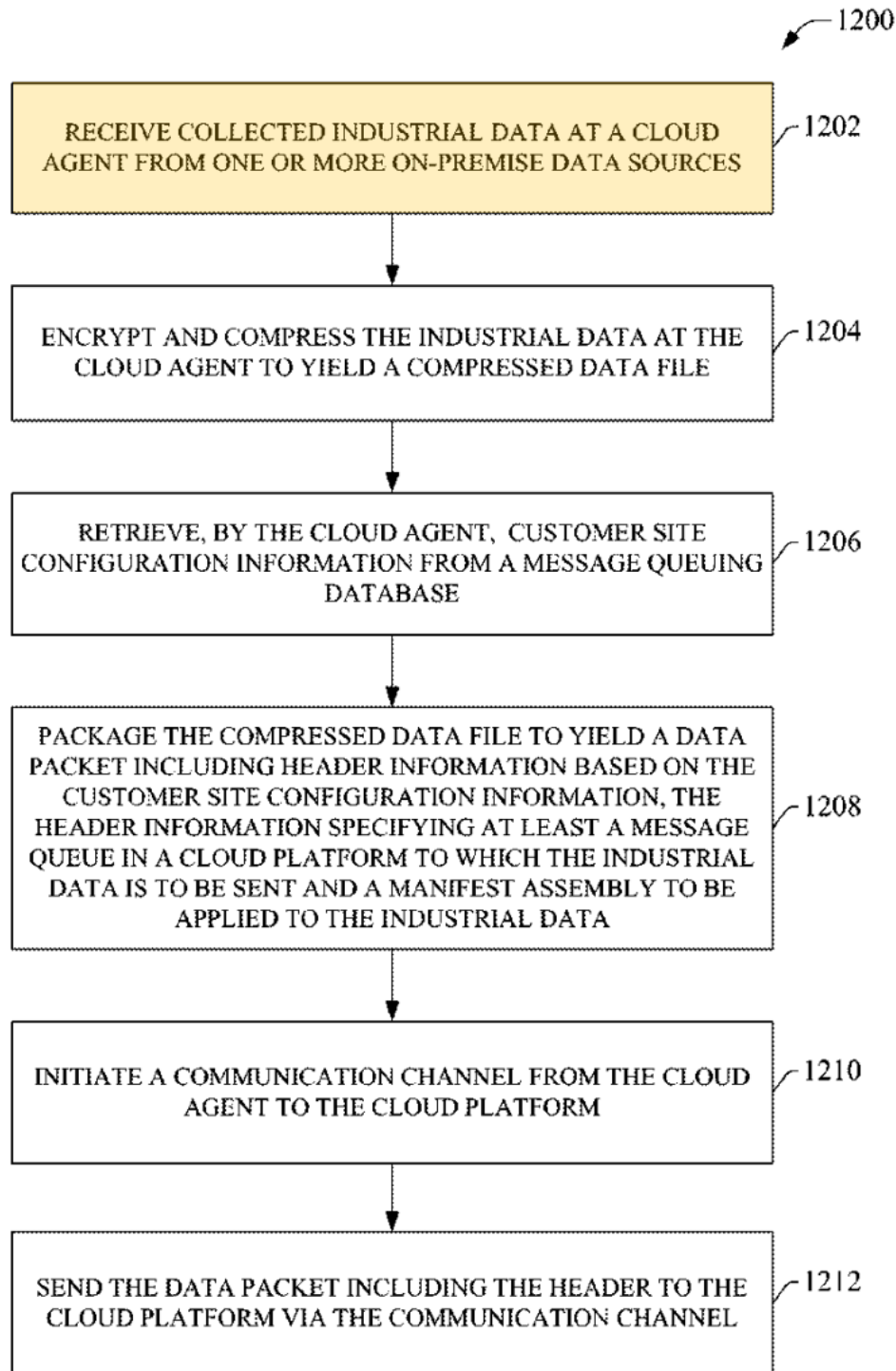


FIG. 12

EX1004, FIG. 12 (highlighting added); EX1003, ¶93.

Regarding a *local network*, Maturana discloses collecting the data on the “plant network.” EX1004, [0048]. Specifically, the cloud agent “collects data items from the various industrial assets on the plant network and packages the collected data according to a generic, uniform data packaging schema used to move the on-premise data to a cloud platform 302.” *Id.* The plant network is a *local network* because the on-premise cloud agent and the industrial controllers are connected “through a physical or wireless *local area network* or radio link.” EX1004, [0040] (emphasis added); *id.*, [0037] (describing plant network); EX1003, ¶94.

Maturana-Kashyap thus discloses and teaches receiving data over a local network from a plurality of computing devices, at least because Maturana expressly discloses doing so. EX1003, ¶95.

6. Element [1.6]

[1.6] apply a plurality of transformations to at least a portion of the received data; and

Maturana-Kashyap satisfies this element. EX1003, ¶96.

Maturana discloses applying a plurality of transformations to at least a portion of the received data. Maturana discloses that the “cloud agent 202 can also perform local analytics on the data prior to moving the data to the cloud platform,” where “[t]his can comprise substantially any type of pre-processing or data refinement that may facilitate efficient transfer of the data to the cloud, prepare the data for enhanced

analysis in the cloud, reduce the amount of cloud storage required to store the data, or other such benefits” (*apply a plurality of transformations to at least a portion of the received data*). EX1004, [0054]; *see also id.*, [0049] (cloud agent “can intelligently sort and organize the data”), [0052] (the cloud agent “packages the compressed data file 412 into a data packet and pushes the data packet to the cloud platform”). EX1003, ¶96.

Maturana expressly teaches *a plurality of transformations* that the cloud agent applies to at least a portion of the received industrial data:

- “[C]loud agent 202 may also *transform a specified subset of the industrial data* from a first format to a second format” in order to “reduce the amount of processing load on the cloud side.” EX1004, [0055] (emphasis added).
- “[C]loud agent 202 may be configured to compress the collected data using any suitable data compression algorithm prior to migrating the data to the cloud platform,” where “[t]his can include detection and deletion of redundant data bits, truncation of precision bits, or other suitable compression operations.” EX1004, [0054].
- “[C]loud agent 202 may be configured to aggregate data by combining related data from multiple sources.” EX1004, [0054].

- “Cloud agent 202 may also encrypt sensitive data prior to upload to the cloud.” EX1004, [0054].
- “[C]loud agent 202 may filter the data according to any specified filtering criterion.” EX1004, [0055].
- “Cloud agent 202 may also associate metadata with selected subsets of the data prior to migration to the cloud, thereby contextualizing the data within the industrial environment.” EX1004, [0056]. This “layered processing of the collected data to generate meta-level knowledge ... can subsequently be leveraged by cloud-based analysis tools to facilitate enhanced analysis of the data in view of a larger plant context.” *Id.*

Each of these example processing steps in Maturana is a transformation. EX1003, ¶97.

Maturana discloses applying more than one such transformation to at least a subset of data by using “also” to describe the different transformations (e.g., “cloud agent 202 may *also* transform” in addition to “compress,” “aggregate,” “encrypt,” and so on). EX1004, [0054]-[0056] (emphasis added). Maturana also expressly discloses applying a plurality of transformations to data in FIG. 12, including encrypting, compressing, and packaging:

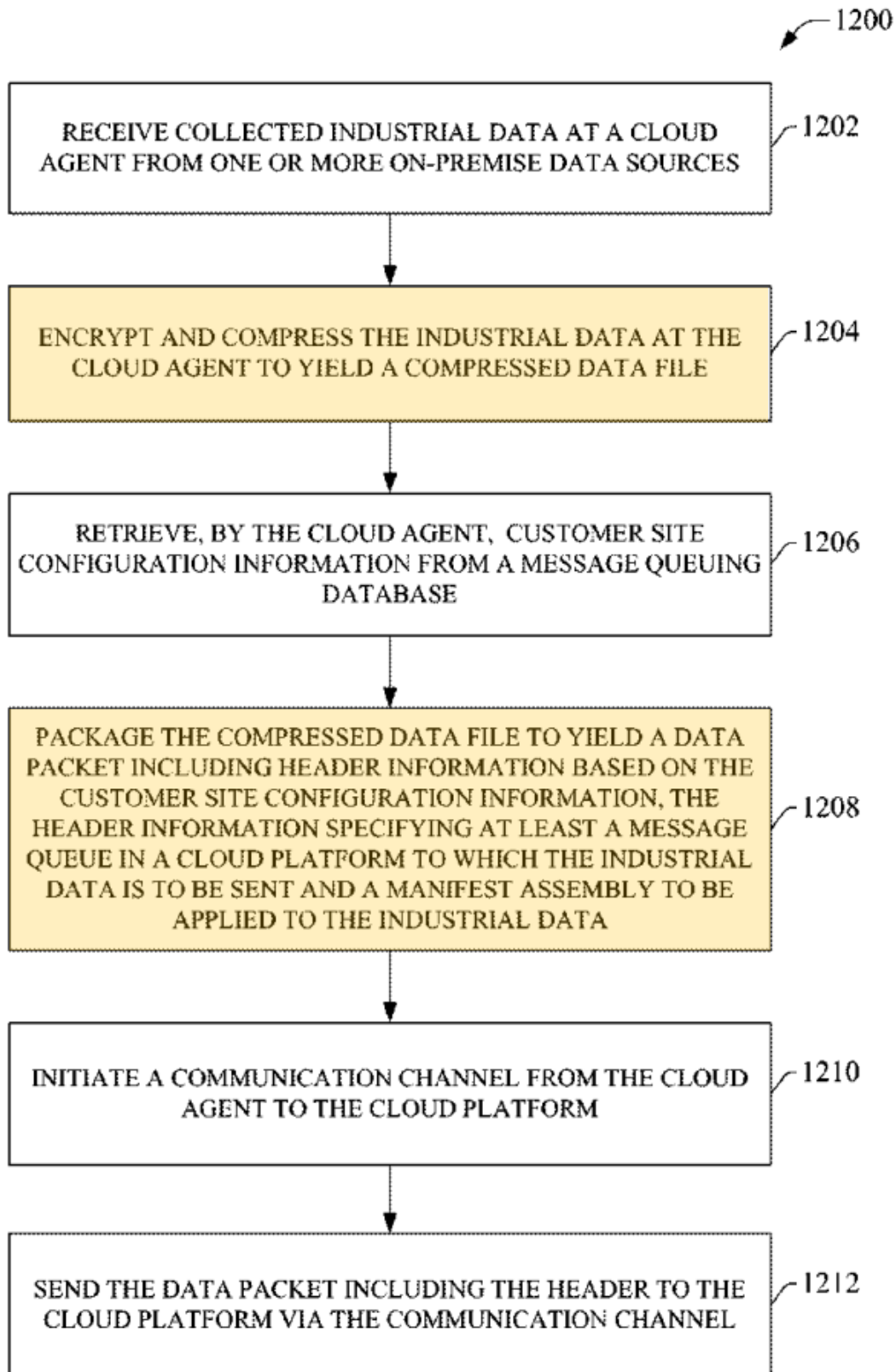


FIG. 12

EX1004, FIG. 12 (highlighting added); *id.*, [0085]. This express disclosure of the cloud agent performing compression, encryption, and data packetization with structured header information in Maturana is express disclosure of *applying a plurality of transformations* because each alters the data in a different and meaningful way: (1) Compression is a transformation because it changes the data's representation and size while preserving content; (2) encryption is a transformation that converts the original data into a different form that cannot be interpreted without a corresponding decryption key, thus altering the data's structure and accessibility; and (3) packaging the compressed and encrypted data into a data packet further transforms the data by embedding it within a structured container that adds metadata governing routing, prioritization, and cloud-side processing. EX1004, [0054], [0069]-[0070], [0085]-[0086]; EX1003, ¶98.

Maturana-Kashyap thus applies a plurality of transformations to at least a portion of the received data, at least because Maturana expressly discloses doing so. EX1003, ¶99.

7. Element [1.7]

[1.7] retransmit the received data over the secure connection as a single data stream.

Maturana-Kashyap satisfies this element. EX1003, ¶100.

Maturana discloses retransmitting the received data over the secure connection as a single data stream. Specifically, the cloud agent “packages the compressed data file 412 into a data packet and pushes the data packet to the cloud platform” (*retransmit the received data over the secure connection as a single data stream*). EX1004, [0052]; *id.*, [0076]. The data is provided as a single data stream because “data from multiple sensors measuring related aspects of an automation system can be identified and *aggregated into a single cloud upload packet* by cloud agent 202.” EX1004, [0054] (emphasis added). EX1003, ¶100.

Thus, Maturana-Kashyap retransmits the received data over the secure connection as a single data stream, at least because Maturana expressly discloses doing so. EX1003, ¶101.

E. Claim 4

1. Element [4.1]

[4.1] A method for ingestion of data into a cloud-based service from an external network, comprising the steps of:

Maturana-Kashyap discloses and teaches this element, and thus renders this claim obvious. *See supra* Section VII.D.1 (element 1.1). Maturana-Kashyap discloses a “methodology for ingesting industrial data to a cloud platform using a cloud agent architecture,” as taught by Maturana. EX1004, [0019], FIG. 12; *id.*,

[0005] (describing “a method for sending industrial data to a cloud platform”); EX1003, ¶102.

2. Element [4.2]

[4.2] installing a midserver comprising at least a processor, a memory, and a plurality of programming instructions stored in the memory and operating on the processor;

Maturana-Kashyap discloses and teaches this element, and thus renders this claim obvious. *See supra* Section VII.D.2 (element 1.2). A POSITA would have understood that the computing device / server executing the cloud agent, as taught by Maturana, is installed at a facility where data is to be collected, and thus the method of ingesting data to a cloud platform taught by Maturana-Kashyap includes *installing* the server at the facility. Maturana discloses the “installation at any facility” (EX1004, [0041]) of a cloud agent executing on a “hardware platform” such as “a server” (*id.*, [0048]). *See also* EX1004, [0026]. Maturana further discusses the physical devices / servers executing the “cloud agents distributed throughout the customer’s industrial enterprise.” EX1004, [0074]. Each server “has a physical address (e.g., a MAC address or other physical address) that uniquely identifies the computing device,” and a given cloud agent is associated with that physical address so that the cloud agent “and the physical device on which the agent executes are logically linked.” *Id.* These disclosures show that a server is *installed* at a particular facility to execute a cloud agent. EX1003, ¶103.

3. Element [4.3]

[4.3] automatically installing, using the midserver, a virtual appliance software application, the virtual appliance software application configured to automatically load a plurality of stored configurations on the midserver;

Maturana-Kashyap discloses and teaches this element, and thus renders this claim obvious. *See supra* Section VII.D.3 (element 1.3). EX1003, ¶104.

4. Element [4.4]

[4.4] establishing a secure network connection from the midserver to an external network;

Maturana-Kashyap discloses and teaches this element, and thus renders this claim obvious. *See supra* Section VII.D.4 (element 1.4). Maturana discloses establishing the secure network connection from the server with the cloud agent (*from the midserver*) to the cloud platform / external network. EX1004, [0057] (“[t]o ensure secure outbound traffic to the cloud”), FIG. 1; EX1003, ¶105.

5. Element [4.5]

[4.5] receiving data over a local network from a plurality of computing devices;

Maturana-Kashyap discloses and teaches this element, and thus renders this claim obvious. *See supra* Section VII.D.5 (element 1.5). EX1003, ¶106.

6. Element [4.6]

[4.6] applying a plurality of transformations to at least a portion of the received data; and

Maturana-Kashyap discloses and teaches this element, and thus renders this claim obvious. *See supra* Section VII.D.6 (element 1.6). EX1003, ¶107.

7. Element [4.7]

[4.7] retransmitting the received data over the secure connection as a single data stream.

Maturana-Kashyap discloses and teaches this element, and thus renders this claim obvious. *See supra* Section VII.D.7 (element 1.7). EX1003, ¶108.

VIII. NO OBJECTIVE INDICIA OF NON-OBVIOUSNESS

Petitioner is unaware of any objective indicia of non-obviousness that overcome the above grounds of unpatentability as to any challenged claim.

IX. CONCLUSION

Petitioner requests institution for claims 1 and 4 on Ground 1 specified in this petition.

Respectfully submitted,

Dated: January 14, 2026

By: /Andrew M. Mason/

Andrew M. Mason, Reg. No. 64,034
andrew.mason@klarquist.com
KLARQUIST SPARKMAN, LLP
One World Trade Center, Suite 1600
121 S.W. Salmon Street
Portland, Oregon 97204

Tel: 503-595-5300
Fax: 503-595-5301

Counsel for Petitioner

**CERTIFICATE OF COMPLIANCE WITH
TYPE-VOLUME LIMITATION PURSUANT TO 37 C.F.R. § 42.24**

This brief complies with the type-volume limitation of 37 C.F.R. § 42.24(a)(1)(i).

The brief contains **9,380** words, excluding the parts of the brief exempted by 37 C.F.R. § 42.24(a).

The brief has been prepared in a proportionally spaced typeface using Microsoft Word for O365 in a 14-point Times New Roman font.

Dated: January 14, 2026

By: /Andrew M. Mason/

Andrew M. Mason, Reg. No. 64,034
andrew.mason@klarquist.com
KLARQUIST SPARKMAN, LLP
One World Trade Center, Suite 1600
121 S.W. Salmon Street
Portland, Oregon 97204
Tel: 503-595-5300
Fax: 503-595-5301

Counsel for Petitioner

CERTIFICATE OF SERVICE
IN COMPLIANCE WITH 37 C.F.R. § 42.6(e)(4)

The undersigned certifies that the **PETITION FOR *INTER PARTES* REVIEW OF U.S. PATENT NO. 11,539,663 and Exhibits 1001 – 1024** were served on January 14, 2026, via **Express Mail** on the Patent Owner at the following address of record as listed on the USPTO Patent Center:

Brian Galvin
Galvin Patent Law LLC
2916 NW Bucklin Hill Road, Suite 485
Silverdale, WA 98383

By: /Andrew M. Mason/
Andrew M. Mason, Reg. No. 64,034
andrew.mason@klarquist.com
KLARQUIST SPARKMAN, LLP
One World Trade Center, Suite 1600
121 S.W. Salmon Street
Portland, Oregon 97204
Tel: 503-595-5300
Fax: 503-595-5301

Counsel for Petitioner